

USER MANUAL

PageGate Version 10 Documentation

© 2022

Contents

Chapter 1	1 Welcome to PageGate version 10
	2 What's New
	3 Release Notes
	7 Purchasing and Pricing
	8 Upgrade
	9 Pricing
	10 Contact Information
	10 Resellers
	10 Evaluation Information
	11 End User License Agreement (EULA)
	16 General Features
Chapter 2	17 How PageGate Accepts Messages
	20 National Weather Service Integration
	24 Accepting HTTP Requests: Webdata API
	29 Controlling Webdata's HTTP Responses
	30 Example: Receiving Replies via EMAG WCTP
	37 API Polling
	38 Example: Spillman XML API
	51 TCP Server Module
	54 UDP Server Module
	57 GetAscii Integration
	58 Command Line Integration
	58 Text File Integration
	59 Registry Polling Integration
	59 How to see if GetAscii is active
	60 How to get a list of Recipients and Groups
	60 Filter Pack
	62 How to Implement the Script
	66 Debugging the Script
	67 The Filter Script

Contents

	92	Complete Script Examples
	135	GetMail Integration
	136	GetWeb Integration
	137	GetSerial Integration
	137	GetTAP Integration
	137	SNPP Server Integration
	139	TouchTone Receiver Integration
	140	Direct Database Access
	141	The Messages Table
	142	The Users and Groups Table
Chapter 3	147	How to Send Messages
	147	SMS (text message) Delivery Methods
	149	Carrier Public Gateways
	151	Cellular Hardware
	152	General SMS Delivery Services
	165	Public Safety Delivery Services
	186	Medical Industry Delivery Services
	193	Carrier Enterprise Systems
	202	Prioritizing Messages
	203	How to send Pages
	204	How to send Email
	205	Using webforms, Web APIs and REST APIs
	206	Templates for HTTP GET Carriers
	210	Templates for HTTP POST Carriers
	215	Miscellaneous Methods
Chapter 4	216	Installation
	218	PageGate Server
	219	Installation Steps
	223	Windows Server 2016 and 2019 Core
	224	PageGate GUI Client
	226	Pre-installation
	226	Installation Steps
	230	Additional PG Admin
	231	Pre-installation

Contents

	231	Installation Steps
	234	Additional PG Monitor
Chapter 5	235	Installation Steps
	239	Running PageGate for the First Time
	240	Standard Setup (MSJet)
	241	SQL Setup
	259	Switching PageGate's Run Style
	262	Starting and Stopping PageGate
	262	Monitoring PageGate
	262	Example Configuration using the PageGate GUI Client
Chapter 6	264	PageGate Admin: How to Configure PageGate
	265	Program
	267	Program Settings
	270	Global Log
	271	Security
	276	Tools
	279	Database
	280	Settings
	281	Status
	282	Tools
	287	Password Protect
	289	Synchronization
	292	Scheduler
	293	Settings
	293	Status
	293	Log
	294	Templates
	296	Template Components
	303	Global Templates
	305	Carrier Templates
	347	Interface Templates
	354	Connector Templates
	357	Recipient Templates
	359	Success and Failure Notification Templates
	361	Interfaces

Contents

362	GetAscii
393	GetMail
407	GetSerial
426	GetTap
431	GetWeb
446	GUI Client
448	Additional APIs
539	Connectors
541	Global
543	Connectors
547	Carriers
550	Prioritizing Messages
551	Creating a new Carrier
585	Context Menu Options
589	Carrier Settings
610	Recipients
611	Prioritizing Messages
613	Creating a Recipient
625	Context Menu Options
633	Recipient Settings
643	Groups
643	Creating a Group
645	Context Menu Options
650	Group Settings
659	Display Groups
660	Creating a Display Group
660	Members
661	Logging
663	Settings
664	Global
666	Scheduler
667	GetAscii
667	GetMail
673	GetSerial
674	GetTap
675	GetWeb
675	GUI Client
676	Connectors

Contents

	678	Audit
	678	Reports
	679	Sent Messages
	690	Pending Messages
	697	Statistics
	704	Carrier List
	705	Recipient List
	706	Group List
	707	Display Group List
	708	Advanced Server and Client Settings
	708	Registry settings for PageGate Server
	711	Registry settings for PageGate Client
	712	Registration
Chapter 7	714	Two Way Messaging and Reply Processing
	715	Two Way Messaging with PageGate's GUI Client
	718	Replies with Cellular Hardware
	719	Static Recipient or Group
	721	Processing replies with GetAscii
	723	Processing replies with a Filter Pack
	732	Replies with APIs and Protocols
Chapter 8	738	Failover Scenarios
	739	Carrier Level Failover
	740	Recipient Level Failover
Chapter 9	741	PageGate GUI Client
	745	MSJet Database Setup (default)
	747	SQL Database Setup
	749	Using the GUI Client
	750	Scheduling Messages
	752	File
	756	Tools
	766	View
	773	Reports
	780	Help
	782	Using a Single GUI Client with Multiple Servers
Chapter 10	783	PageGate Monitor

Contents

	784	File
	785	Settings
	787	View
	788	Tools
	789	Help
	789	Tips
	790	About
Chapter 11	790	Keeping an eye on PageGate
	790	PageGate Log Monitor Feature
	792	PG Monitor Alerts through NotePager Pro
Chapter 12	794	Support

Welcome to PageGate version 10

1

At its most basic level, PageGate is a communication program that translates inputs to turn them in to outputs, usually in the form of messages. Put another way, PageGate allows you to receive and collect data in a variety of formats to deliver as SMS, email, and pages, among many other formats to your recipients.

Beyond that basic explanation, however, PageGate is a sophisticated messaging application. The heart of the program is a high throughput messaging engine that has the ability to support multiple connectors that can send SMS, pages and Email as well as more complex forms of communication such as GET/POST to both request data from or push data to an API.

PageGate supports several front-end interfaces (APIs) that can be used alone or in conjunction with each other. This allows PageGate to receive messages from many sources simultaneously, and deliver them to recipients such as cell phones, alphanumeric pagers and email addresses, among many others. These APIs tell PageGate who a message needs to go to, who or what sent the message, what the message should say and when the message should be sent.

For example, PageGate's GUI Client API is a program that can be installed on any workstation that has UNC/network access to the PageGate server's shared database. This provides a graphical messaging interface for users on workstations and can be incredibly useful for dispatching messages.

As another example, PageGate's GetASCII API allows the program to monitor a directory or series of directories on the local system or a network path for text/ASCII formatted files. The module also provides a command line executable for other programs to pass messaging commands to. This module is used by network performance monitoring programs such as SolarWinds Network Performance Monitor/Orion and Paessler's PRTG and is also used by many computer automated dispatch platforms as well.

For more information on all of PageGate's APIs, have a look at the [Accepting Messages](#) section of the documentation.

Once PageGate knows that a message needs to be delivered, it supports many delivery protocols for sending messages to many devices and receiving systems. It can send messages to cell phones, alphanumeric pagers, email addresses, web servers, PIMs and chat clients, among others.

For more information on how PageGate can deliver messages, have a look at the [Sending Messages](#) section of the manual.

As an example of a simple configuration, PageGate can accept messages from a dispatcher, who manually enters messages into PageGate's GUI Client application to send those messages to cell phones, pagers and email addresses.

In a more complex example, PageGate can be simultaneously receiving hundreds of messages from multiple sources such as email, webpages, monitored serial ports for alarm systems, manual dispatchers and integration with other applications while also delivering them through up to sixteen concurrent outbound communication sessions implementing multiple protocols to multiple sources.

PageGate can be installed to run on a single non-dedicated workstation, or using its modular architecture, PageGate can actually be installed across a dozen or more computers for a truly high-performance messaging gateway system.

What's New

Take advantage of the powerful and efficient PageGate version 10!

- **SQL Database Module**

PageGate now supports using SQL Server and SQL Server Express, allowing you to integrate PageGate into an existing SQL Server environment or use a dedicated instance of SQL Server Express! Of course the existing embedded Jet database engine is still supported.

- **Replies and two-way messaging with GUI Client**

PageGate's GUI Client now supports receiving messages and sending replies to those messages! This allows your recipients to reply to messages and have those replies go to instances of the GUI Client. GUI Client users can also move those replies between each other, so the replies can be handled by the person best suited to do so. Additionally, workstations can send messages to each other in a chat style conversation.

- **Secure Deployment with SQL**

Deploying PageGate with an encrypted SQL database connection allows you to guarantee that any database connection to the server is secure.

- **Secure Client Connection to Cloud PageGate Server with SQL**

Deploy PageGate to a cloud server with an encrypted SQL connection and you can offer your personnel the ability to use the PageGate Client securely from anywhere!

- **National Weather System Integration**

PageGate now supports monitoring the National Weather System's alert, watch and warning system to automatically notify your recipients with essential information!

- **Scheduled and Repeating Messages with PageGate Admin**

PageGate Admin now has the ability to schedule messages to go out in the future and can also schedule repeating messages, making it much easier to create routines to poll data from APIs.

- **Enhanced Internet API Support**

In version 10, we've improved PageGate's ability to bi-directionally communicate with nearly any web based API. This allows you to both send information to and receive information from many Internet and intranet sources.

- **Enhanced support for cellular hardware and infrastructure**

PageGate version 10 has continued to build on previous versions' hardware support and can use any style of [cellular hardware](#) you may need to interact with, from cellular gateways on virtual machines to cellular modems connected to physical servers.

- **Improved global performance**

Major work has been done to the core of PageGate for version 10, drastically improving its efficiency and usage of resources, like memory, CPU, and database. The result is a noticeable increase in its operational speed.

- **Productivity**

Numerous productivity improvements have been made to PG Admin and PG GUI Client, including a brand new search function!

- **Improved Data Handling**

Speed, and scope improvements have been made to PageGate's import, export, backup, logging, and synchronization features.

- **Look and Feel**

The look and feel of the GUI Client, PG Admin and PG Monitor programs have been updated to have a more modern appearance.

- **Filter Pack**

The optional Filter Pack has a number of added capabilities and improvements, allowing for even more flexibility and support for nearly any data format. This is especially useful when interacting with internet and intranet APIs!

Release Notes

Changes since 9.0.2

- Option for direct MS SQL Server database support, instead of embedded Jet database or linked SQL database
- Extensive optimizations for more efficient use of processor, database, and memory resources
- Updated look and feel of PG Admin, PG Monitor, and PG Client (added toolbar to PG Client)
- New local message and reply handling added to PG GUI Client

Welcome to PageGate version 10

1

- Huge performance improvements made to import and synchronization routines
- Subnet values can be added to BlockedIPs and AllowedIps files for mail server
(xxx.xxx.xxx.xxx/xx)
- Added Database Export All and Import All options in PG Admin
- Added Database backup prompting to PG Admin
- (use No_Admin_Backup_Prompt - REG_SZ = True, to disable on single computer)
- In PG Admin added right-click popup 'Open Folder' options for Program, Database, and Logging
- Double-clicking on logs in PG Monitor will open the log in notepad
- Added StripNonAscii setting to TCP and UDP Servers
- Added template variables: FilterDebugFile, SubjectFilterDebugFile, InterfaceFilterDebugFile,
- RawInterfaceFilterDebugFile, ReplyFilter, ReplyFilterDebugFile, Unique, LongUnique,
- *note /append switch can be used after a XFilterDebugFile filename
- Added template variable: UseFilterDll
- Added SenderIps setting to TAP IP Server, SNPP Server, TCP Server, and UDP Server modules
- HTTPGET and HTTPPOST protocols - added HTTPResponseTimeout template setting
- RawTCP protocol - added RawTCPSuccessResponse, RawTCPErrorResponse, RawTCPResponseTimeout,
- ResponsePath, ResponseRecipient
- RawSerial protocol - added RawSerialSuccessResponse, RawSerialErrorResponse, RawSerialResponseTimeout,
- ResponsePath, ResponseRecipient
- *** Due to the response timeout now being able to be set for the above protocols, the default
- timeout value has been lowered.
- Added new template Setting: RawTCPProtocol=ssh
- Added new template setting: HttpProtocol=https
- Added new template settings for GSM protocols: GsmUseCpoint, GsmIpsProtocol=ssh

Welcome to PageGate version 10

1

- In PG GUI Client, the pre-programmed message buttons grow with form resize
- Added Interface templates to export/import routines
- Added support for multiple dedicated Carriers for each Connector (separated by |)
- Added WaitForCr to TAP IP Server
- Improved UrlEncoding in HTTPGET, HTTPPOST, and AwsSns protocols
- Added ParameterDelimiter template variable
- Increased max message size to 16000 characters
- Separated SNPP Advanced screen Password into Username and Password fields
- - using username <space> password in the password field still works
- Increased valid port numbers for SMTP protocol to > 32768
- Clicking on module name in PG monitor brings its monitor window to the front
- New replacement webgate.exe CGI module
- Added SmtxFriendlyNames template setting
- Added ArchiveLog feature to SNPP, TAP-IP, TCP and UDP servers
- Added Delete as a valid HttpPostType template variable value
- Added ResponseIncludeRequest template setting
- Added service startup error logging to Windows event logs
- Optimized start, stop, restart, and cleanup routines
- Added GetSerial_Ignore_Restart, GetTap_Ignore_Restart registry settings
- Added 0 and -1 as valid values to the Database "Purge After (days)" setting
- Added filter pack support to SNPP, TCP, UDP and TAP-IP servers
- Added warning message if a recipient is added that will exceed the recipient license
- In PG Admin added 'SQL' object under 'Database' in the tree - for setting SQL Server connection strings
- Added Tools->Reset Counter to GUI Client
- Added support for headers in HTTPGET protocol
- Added right-click 'Scheduled' messages options for carriers, recipients, and groups
- Added logging support in PG Monitor for multiple tcp, udp, tap-ip and snpp server modules
- Added option in PG Admin to schedule messages

Welcome to PageGate version 10

1

- Database synchronization feature now syncs oncall tables, ad-hoc maps, mailboxes and templates
- Import/Export of groups has now been split into separate groups and display groups options
- Updated order of operations for StripCharacters when processing templates
- Added SubjectMaxChars and BodyMaxChars to template settings
- Renamed RawTcpProtocol and GsmIpProtocol template settings to TcpProtocol
- Added 'Find' function in PG Admin (ctrl-f)
- Improved PG Client startup time
- In PG Admin, hovering over a recipient or group now shows its full name
- An Upgrade Summary file is written to the database folder with any registry and template settings
- changes made during the upgrade - SettingsUpgradeReport.txt
- PGFilter:
 - - improved debug logging - added many new debug options
 - - added new MatchFound, MatchNotFound, Lookup, ReadFile, Run, RunFilter and Goto sections
 - - added commandline support for debug settings
 - - added .dll version (better performance than existing .exe version)
- Changed default SMTP content transfer encoding from quoted-printable to 7bit
- Renamed remaining references to 'Dialers' to 'Connectors'
- snppserver.exe and snpp.ini renamed to pgsnppserver.exe and pgsnpp.ini
- tapipserver.exe and tapip.ini renamed to pgtapipserver.exe and pgtapip.ini
- ID in recipient table now takes preference over ID in message table, for all non-ad-hoc messages
- Fixed problem where PG Client sometimes didn't get removed from task list
- Fixed problem with registration screen showing every time PG Admin is started, on some systems
- Fixed PG Monitor jumpy scrolling at times

Welcome to PageGate version 10

1

- Fixed problem where PG Admin didn't immediately show templates for imported carriers and recipients
- Fixed SMTP date format
- Fixed time zone offset problem when using SMTP and port 25 (may still have issues in Win XP)
- Fixed possible race condition in 'resetting database flags' section of cleanup routine

Purchasing and Pricing

For a quote, please contact NotePage Sales at 781-829-0500 or sales@notepage.com

To purchase PageGate, you have several options:

- Online (with credit cards: MasterCard, Visa, American Express and Discover) at <https://www.notepage.net/purchase.htm>
- Fax credit card details (i.e. number, expiration date, security code, name and billing address) and order information to NotePage, Inc. at 781-829-0419.
- Purchase order, signed and dated, on official letterhead can be faxed to NotePage, Inc. at 781-829-0419.
We are unfortunately unable to accept purchase orders from outside of the USA. Obtain W-9 at <https://www.notepage.net/purchase-orders.htm>

- Check and money order (in US Dollars, drawn from a US bank) payable to:

NotePage, Inc.

P.O. Box 296

Hanover, MA 02339

USA

- For a wire transfer or ACH payment, please contact NotePage Sales at sales@notepage.com for bank and account details
- NotePage Reseller listed at <https://www.notepage.net/reseller.htm>
For more NotePage Resellers, please contact NotePage Sales Department at 781-829-0500 or sales@notepage.com

When purchasing PageGate, please include the following order information:

1. PageGate Recipient License quantity.

Welcome to PageGate version 10

1

2. Interface(s) & optional Additional Connectors.
3. Mailing address and attention to for the CD.
4. Contact name, email and telephone number.
5. Email address for Registration Key delivery.
6. Software registration information includes these fields:

Name/Department:

Company Name:

Address1:

Address2:

City:

State/Province:

Zip Code/Postal Code:

Country:

A PageGate electronic Registration Key will be emailed within 24 hours of ordering. A PageGate CD will be sent by regular mail.

For information on pricing for PageGate, please call our Sales Department at 781-829-0500 xt 1. Alternatively, you can email us at sales@notepage.com or visit our website, <https://www.notepage.net>

Upgrade

1. Online (with credit cards: MasterCard, Visa, American Express and Discover) at <https://www.notepage.net/upgrades.htm>

2. Fax credit card details (i.e. number, expired date, security code, name and billing address) and order information to NotePage, Inc. at 781-829-0419.

3. Purchase order, signed and dated, on official letterhead can be faxed to NotePage, Inc. at 781-829-0419. We are unfortunately unable to accept purchase orders from outside of the USA. Obtain W-9 at <https://www.notepage.net/purchase-orders.htm>

For a quote, please contact NotePage Sales at 781-829-0500 or sales@notepage.com

4. Check and money order (in US Dollars, drawn from a US bank) payable to:
NotePage, Inc.
P.O. Box 296
Hanover, MA 02339
USA

5. For a wire transfer, please contact NotePage Sales at sales@notepage.com

6. For a NotePage Reseller in your area, please contact NotePage Sales.

Please include the following upgrade information:

1. PageGate Serial Number (PageGate Admin module under Registration).
2. PageGate existing version to current version and PageGate CD (optional).
3. PageGate Pager License: existing quantity to new quantity.
4. Interfaces: GUI Client, Command Line/ASCII, Web (CGI), TAP, TAP2ASCII Module, Serial, Serial2ASCII Module, SNPP Server Module, Tap IP Server Module, Touchtone Receiver Module (DTMF), PageGate Filter Pack, Android SMS Gateway App
5. Additional Connectors.
6. Contact name, email and telephone number.
7. Mailing address and 'attention to' for the optional PageGate CD.
8. Email address for Registration Key delivery.

An upgraded PageGate electronic Registration Key is emailed within 24 hours of ordering. The optional PageGate CD, if ordered, is sent by regular mail. The optional PageGate CD is only relevant when upgrading the PageGate version.

Pricing

PageGate's pricing is determined by three things and the option of a service license agreement or standard support.

• **Recipients**

Recipients can be cell phones, email addresses, web APIs, pagers, fax machines and similar. This represents the people, devices, APIs and other destinations for your messages.

• **Interfaces**

These represent how you want PageGate to accept messages for delivery. PageGate's Interfaces are priced separately.

• **Connectors**

These represent the throughput of the system, the rate at which the program can deliver messages. PageGate includes 1 Connector. Additional Connectors are a separate, one-time charge.

There are no monthly or yearly fees for PageGate.

Specific pricing can be found at <https://www.notepage.net/purchase.htm>

Welcome to PageGate version 10

• **PageGate Priority Support**

Priority support puts your inquiries in a priority queue for a guaranteed 2-hour response during normal business hours and also provides upgrades to PageGate free of charge for the duration of the contract.

For more information, see <https://www.notepage.net/priority-support.htm>

Contact Information

NotePage, Inc.
PO Box 296
Hanover, MA 02339
USA

Sales: sales@notepage.net

Support: [Please see the 'Support' section of the documentation](#)

Phone Number: 781-829-0500

Fax Number: 781-829-0419

Websites: www.notepage.net

Resellers

More information at <https://www.notepage.net/reseller.htm>

For NotePage's Reseller programs please contact NotePage, Inc. at 781-829-0500 or sales@notepage.com

Evaluation Information

PageGate software defaults into a evaluation mode until purchased and registered. Enter the provided registration information in the PageGate Admin module's Registration screen.

The evaluation version of PageGate allows for two active recipients at any given time, and the PageGate Server and PageGate will need to be restarted every 3-5 days.

If a less restrictive evaluation would be useful to fully test PageGate, please contact NotePage Sales at sales@notepage.com or by phone at 781-829-0500 x 1 to discuss the availability of a thirty-day temporary evaluation key.

This software is not free. Using an unregistered copy of NotePage, Inc.'s software outside the trial period is considered software piracy and is illegal. The registration

Welcome to PageGate version 10

1

fee allows us to continue developing quality products, notify you of updates, and provide technical support. PageGate can be tried and tested for an unconditional trial.

If you like PageGate and want a complete, registered version please contact: sales@notepage.com

If there are any questions, please contact sales@notepage.com

End User License Agreement (EULA)

END-USER LICENSE AGREEMENT for PageGate

IMPORTANT PLEASE READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE CONTINUING WITH THIS PROGRAM INSTALL:

NotePage, Inc.'s End-User License Agreement ("EULA" or "AGREEMENT") is a binding legal agreement between you ("END-USER") (either an individual or a single entity) and NotePage, Inc. for the NotePage software product(s) identified above which may include associated software components, media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. This license agreement represents the entire agreement concerning the program between you and NotePage, Inc., (referred to as "licensor"), and it supersedes any prior proposal, representation, or understanding between the parties. If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT.

By downloading/installing or using the SOFTWARE PRODUCT, you

- (a) acknowledge that you have read and understand this AGREEMENT;
- (b) represent that you are 18 years of age or older; and
- (c) accept this AGREEMENT and agree that you are legally bound by its terms. If you do not agree to these terms. Do not download, install or use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE.

Welcome to PageGate version 10

1

The SOFTWARE PRODUCT is licensed as follows:

(a) Installation and Use.

NotePage, Inc. grants you the right to install and use copies of the SOFTWARE PRODUCT on your computer running a validly licensed copy of the operating system for which the SOFTWARE PRODUCT was designed [e.g., Windows 95®, Windows NT®, Windows 98®, Windows ME®, Windows XP®, Windows 2000®, Windows 2010®].

(b) Non-Transferable.

The SOFTWARE PRODUCT license is non-transferable.

(c) Proper Licensing.

A single SOFTWARE PRODUCT license is for a single installation, at a single location, if you wish to run the software at multiple locations and multiple instances of the server software, you must purchase additional SOFTWARE PRODUCT license(s).

(d) Backup Copies.

You may also make copies of the SOFTWARE PRODUCT as may be required by law and necessary for backup and archival purposes.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

(a) Maintenance of Copyright Notices.

You must not remove, delete, alter or obscure any copyright notices or trademarks on any and all copies of the SOFTWARE PRODUCT.

(b) Distribution.

You may not distribute registered copies of the SOFTWARE PRODUCT to third parties. Evaluation versions are available for download from NotePage, Inc.'s websites may be freely distributed.

(c) Prohibition on Reverse Engineering, Decompilation, and Disassembly.

You may not reverse engineer, decompile, disassemble, decode or otherwise attempt to derive or gain access to the source code of the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

(d) Rental or Assignment.

You may not rent, lease, lend, sublicense, transfer or assign the SOFTWARE PRODUCT.

(e) Modifications

Welcome to PageGate version 10

1

You may not modify, translate, adapt or otherwise create derivative works or improvements of the SOFTWARE PRODUCT.

(f) Security or Licensing

You may not remove, disable, circumvent or otherwise create or implement any workaround to any copy protection, rights management, licensing or security features in the SOFTWARE PRODUCT.

(g) Support Services.

NotePage, Inc. may provide you with support services related to the SOFTWARE PRODUCT ("Support Services"). Any supplemental software code provided to you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this EULA.

(h) Compliance with Applicable Laws.

You must comply with all applicable laws regarding use of the SOFTWARE PRODUCT.

3. TERM AND TERMINATION

Without prejudice to any other rights, NotePage may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT in your possession.

(a) The term of this agreement commences when you download/install the SOFTWARE PRODUCT

(b) The END-USER may terminate this agreement by deleting the SOFTWARE PRODUCT and all copies from all computers.

(c) NotePage may terminate this AGREEMENT at any time without notice if it ceases to support the SOFTWARE PRODUCT, which NotePage may do at its sole discretion.

(d) NotePage may terminate this AGREEMENT with or without notice if the END-USER violates any of the terms and conditions of this AGREEMENT.

(e) NotePage reserves the right to terminate support to any END-USERS that violate the terms of this AGREEMENT.

(f) Upon Termination

(i) all rights granted to you under this AGREEMENT will also terminate

(ii) the END-USER must cease all use of the SOFTWARE PRODUCT and delete all copies of the SOFTWARE PRODUCT.

(g) Termination will not limit any of NotePage's rights or remedies.

4. COPYRIGHT

All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by NotePage, Inc. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by NotePage, Inc.

5. THIRD PARTY COMPONENTS

The SOFTWARE PRODUCT may contain third-party components. NotePage does not assume and will not have any liability or responsibility to any entity for any third party components.

6. NO WARRANTIES

NotePage, Inc. expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT is provided "As Is" without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, noninfringement, or fitness of a particular purpose. NotePage does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the SOFTWARE PRODUCT. NotePage, Inc. does not have any liability arising from or related to the END-USERS use of or inability to use the SOFTWARE PRODUCT. NotePage makes no warranties respecting any harm that may be caused by the transmission of a computer virus, worm, time bomb, logic bomb, or other such computer program. NotePage further expressly disclaims any warranty or representation to Authorized Users or to any third party.

7. LIMITATION OF LIABILITY

In no event shall NotePage, Inc. be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) rising out of 'Authorized Users' use of or inability to use the SOFTWARE PRODUCT, even if NotePage, Inc. has been advised of the possibility of such damages. In no event will NotePage be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or

otherwise. NotePage, Inc. shall have no liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, moral rights or the disclosure of confidential information. NotePage, Inc. shall not have any liability for any personal injury, property damage, lost profits, cost of substitute goods or services, loss or corruption of data, loss of goodwill, business interruption, computer failure or malfunction or any other consequential, incidental, indirect, exemplary, special or punitive damages that results from the use or inability to use the SOFTWARE PRODUCT.

8. INDEMNIFICATION

The END-USER agrees to indemnify, defend and hold harmless NotePage, Inc. and its officers, directors, employees, agents, affiliates, service providers, successors, and assigns from and against any and all losses, damages, liabilities, deficiencies, claims, actions, judgements, settlements, interest, awards, penalties, fines, costs or expenses of whatever kind, including reasonable attorney's fees, arising from or relating to the END-USERS use or misuse of the SOFTWARE PRODUCT or the END-USERS breach of this AGREEMENT.

9. UPGRADES

NotePage may from time to time in its sole discretion develop and release upgrades to the SOFTWARE PRODUCT, the upgrades, may include updates, upgrades, bug fixes, patches and other error corrections and/or new features (collectively including related documentation). Upgrades may also modify or delete in their entirety certain features and functionality. NotePage has no obligation to provide upgrades or to continue to provide or enable any particular features or functionality. NotePage may charge for upgrades. All Upgrades will be deemed part of the SOFTWARE PRODUCT and may be subject to all terms and conditions of this AGREEMENT.

10. REVISIONS TO EULA

NotePage may revise this EULA at any time by posting an updated version, although we will endeavor to provide you with prior notice of any material changes to this EULA. Continued use of the SOFTWARE PRODUCT after a change to the EULA constitutes your binding acceptance of this EULA.

11. GOVERNING LAW

This Agreement is governed by and construed in accordance with the laws of the State of Massachusetts. Any legal suit, action or proceedings arising out of or related to this Agreement and to the SOFTWARE PRODUCT must be instituted exclusively in the State of Massachusetts.

12. ENTIRE AGREEMENT

The AGREEMENT constitutes the entire AGREEMENT between the END-USER and NotePage, Inc. with respect to the SOFTWARE PRODUCT and supersedes all prior or contemporaneous understandings and AGREEMENTS, whether written or oral, with respect thereto. If any provision of this AGREEMENT is illegal or unenforceable under applicable law, the remainder of the provision will be amended to achieve as closely as possible effect of the original term and all other provisions of this AGREEMENT will continue in full force and effect. Liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, moral rights or the disclosure of confidential information.

General Features

- **Multiple recipients** - allows a single message to be sent to more than one recipient and/or group.
- **Recipient Groups** - messages can be sent to predefined groups of recipients.
- **Display Groups** - dispatchers using the PageGate GUI Client program can be configured to only see a subset of recipients and groups
- **On-call groups** - a schedule can be setup for the recipients in a group, allowing PageGate to send messages to only the recipients scheduled to receive messages at that point in time.
- **Multiple delivery methods** - PageGate can be connected to the internet through a modem, and a paging terminal by a direct cable connection.
- **Message splitting** - long messages can be intelligently split into multiple smaller messages.

- **Settable character limits** - the maximum number of characters to be sent in a message is settable by carrier and individual recipient
 - **Modular / Scalable** - PageGate's front-end Interfaces allows the program to be used with network computers, email, websites and 3rd party applications.
 - **Protocols** - for message delivery PageGate supports the protocols: SNPP, SMTP, WCTP, HTTPS, TAP, FAX, GSM, UCP and TONE.
 - **Requeue messages** - missed messages can be requeued for re-transmission to the wireless carrier.
 - **High output** - PageGate's intelligent queuing of messages and multiple Connectors allows for the delivery of a large number of messages.
 - **Failover** - undeliverable text messages can be automatically sent to an alternative carrier, recipient, group and connection.
 - **Intelligent message queuing** - messages for the same carrier are grouped together and sent during a single communication session.
 - **Maximum Characters** - set the number of characters for each recipient and carrier. When sending a larger amount of characters the messages will be sequentially numbered.
 - **Reliepage** - will number a message and a recipient can tell if a sequentially numbered message is missed.
 - **Hold Messages** - for recipients and groups to be sent at a later time.
 - **Reports** - on recipient, date/time and message can be previewed, saved and printed for all messages sent and received.
 - **Logs** - to help monitor the activity of all of PageGate modules and functions.
 - **Real-time Monitoring** - with the PageGate Monitor module of all PageGate's modules and functions from the paging server, or any workstation connected via TCP/IP.
- PageGate is designed to easily integrate with other applications by providing a range of front-end APIs. Once integrated, your application(s) will be able to send messages to the recipients and groups that have been configured in the software, and even send messages through PageGate for recipients that aren't configured in the program's database with PageGate's Ad-Hoc Messaging feature.

How PageGate Accepts Messages

2

PageGate supports the following integration methods:

- [National Weather Service Integration](#)

PageGate can monitor the National Weather Service's alerts, warnings and watches system to automatically send notifications when severe weather events happen.

- [GUI Client](#)

The PageGate GUI Client is a Windows based program that interacts with PageGate's database. [Recipients](#) and [groups](#) configured in the [PageGate Admin](#) display in the recipient list, which allows a user to select them to send a message immediately or schedule a message to go out at a later time.

When using an MSJet database (default), the PageGate Client will need access to the PageGate server's database directory. You will need to share that folder on the server and provide all appropriate users and usergroups the read, write and modify permissions.

When using an SQL database, the PageGate Client will need a configuration file that tells it how to interact with the SQL server.

When using an SQL database hosted on a cloud server, if you enable 'Always Encrypted' in SQL, GUI Client application can be installed on any computer and given access to your PageGate server. This makes connecting remote workstations far easier as they can be given direct access.

- [Accepting HTTP Requests](#)

PageGate has an HTTP data receiver (webdata.exe) that can receive input from nearly any API or protocol that offers an HTTP return like HTTP GET, HTTP POST and WCTP, which allows you to receive JSON, XML, HTML, WCTP and similar data submissions via HTTP/S request.

- [API Polling](#)

This method allows PageGate to request information from an API using a webform, HTTP GET or HTTP POST method. The information collected from the APIs is then fed through the GetASCII interface and filter scripts for parsing and, as a result, require the GetASCII interface and, optionally, filter pack.

- [TCP Server Module](#)

This module allows PageGate to run a TCP server/listener, enabling it to accept TCP packets from any source.

- [Syslog/UDP Server Module](#)

This module allows PageGate to run a UDP server/listener, enabling it to accept UDP packets and function as a Syslog receiver.

- [GetASCII - Command Line/ASCII Interface](#)

This interface is capable of monitoring a directory or series of directories on the local hard drive or across the network for ascii file output and also provides a command line interface for other applications to use. This module allows many 'off the shelf' and custom applications, such as SolarWinds' Network Performance Monitor, SolarWinds' Orion, Paessler's PRTG and several CAD programs, among many others, to tell PageGate when a message needs to be sent.

- [Filter Pack](#)

This is a companion module for GetASCII and allows PageGate to conditionally modify nearly anything about any part of any message, whether they're received by one of the interfaces or being sent by one of the connectors.

- [GetMail - Email Interface \(Includes SMTP server\)](#)

This API unlocks the ability to receive email in to PageGate by enabling the receiving functions of PageGate's mail server engine. This API provides the ability to host your own messaging domain or sub-domain as well as unlocking an email forwarding feature associated with your recipients and groups. This API also allows PageGate to function as an SMTP/Email receiver for any installation of [NotePager Pro](#), allowing satellite offices to submit messages for processing with a centralized, master server.

- [GetWeb - Webserver and CGI Interface](#)

Also known as the GetWeb Interface, this API allows PageGate to integrate with a web server to provide a website messaging interface. This allows you to host messaging websites on your local intranet or, if GetWeb is interfaced with an outward facing web server, on the internet. Web Developers can also use this web server interface to send GET/POST requests to notify PageGate when a message needs to be sent, who the message should go to, who sent the message and what the message should say. This interface can also be used to enable PageGate to receive notifications from applications running on different platforms (Unix, Mac, etc.).

- [GetSerial - Serial Port \(physical or virtual\) Interface](#)

Also known as the GetSerial Interface, this API allows PageGate to monitor a serial/RS232 port (physical or virtual) for raw data ASCII strings to be parsed in to outbound messages. When a data string is sent to a monitored serial port, GetSerial pattern matches sections of the data string to parse information for messaging.

- [Serial to ASCII Module](#)

This is a companion module for GetSerial and unlocks the ability to monitor additional serial/RS232 ports (physical or virtual), up to 16. This module requires the GetAscii API to function.

- [GetTAP - TAP Interface](#)

How PageGate Accepts Messages

2

Also known as the GetTAP Interface, this API allows PageGate to monitor a serial port or answer a modem call to receive TAP transmitted pages. This allows PageGate to function as a TAP receiver or a TAP terminal for any devices and systems that can transmit that protocol, such as a Zetron or Motorola paging terminal as well as a few CAD systems. This also allows PageGate to function as a TAP receiver for any installation of [NotePager Pro](#), allowing satellite offices to submit messages for processing with a centralized, master server.

- [TAP to ASCII Module](#)

This is a companion module for GetTAP and unlocks the ability to monitor additional serial/RS232 ports or modems, up to 16. This module requires the GetAscii API to function.

- [TAP over TCP/IP Server Module](#)

This API allows you to receive the TAP protocol by TCP/IP instead of by serial or modem line. This module requires the GetAscii API to function.

- [SNPP Server Module](#)

This API allows PageGate to function as an SNPP server to receive this protocol from any source that can send it. Messages received by this API are output to a directory as a text file that is then read in with the GetAscii API. This also allows PageGate to function as an SNPP receiver for any installation of [NotePager Pro](#), allowing satellite offices to submit messages for processing with a centralized, master server. This module requires the GetAscii API to function.

- [TouchTone Server Module \(DTMF\)](#)

This API provides the ability to monitor a voice modem to answer calls and receive DTMF signals and output them as text to a file for the GetAscii API to process. This module requires the GetAscii API to function.

- [Registry Polling](#)

The GetAscii API can accept messages that are passed to it via the Windows registry.

- [Direct Database Access](#)

PageGate uses an MSJet database that can be linked to other databases by ODBC expression.

National Weather Service Integration

Requirements: GetAscii, Filter Pack

The NWS (National Weather Service) provides a live feed of public alerts based on state, zone and county at their website <https://alerts.weather.gov/>

PageGate can be configured to routinely poll the NWS for weather alerts in any area of interest and automatically trigger notifications, alerts and messages based on them.

To start, we'll need to create a few directories. While they can reside anywhere, you'll need to create:

```
\weather\  
\weather\posts\  
\weather\ids\
```

For example, if you're using PageGate's default data location, we would recommend creating:

```
c:\PageGateData\Weather\  
c:\PageGateData\Weather\posts\  
c:\PageGateData\Weather\ids\
```

Next, we'll need to configure the National Weather Service API in PageGate:

1) Open PG Admin.

2) Right click Carriers, select 'Add'.

3) Specify the following:

Name: NWS-GetData

Protocol: HTTPGET

Host: The link to the NWS data feed. See note below

Port: 443

Max Chars: 16000

NOTE: The 'Host' is the link to the NWS data feed you're interested in. To find that link, go to <https://alerts.weather.gov/> and scroll to the Zone and County list. When you find the state, zone or county you're interested in, right click on the link for the specific Zone or County code and copy it.

After pasting the URL into the 'Host' field, remove the `https://`

For example, the link to the NWS data feed for Alameda, CA is <https://alerts.weather.gov/cap/wwaatmget.php?x=CAC001&y=1>

We would need this in 'Host' field:

```
alerts.weather.gov/cap/wwaatmget.php?x=CAC001&y=1
```

4) Click Apply.

5) Go to Carriers - NWS-GetData - Template

6) In the 'Settings' section of the Template, specify the following:

ResponsePath = The path to the weather folder. Note: Declare \ as \092

Example:

How PageGate Accepts Messages

2

ResponsePath = C:\092PageGateData\092weather

7) Click Apply.

Next, we'll need to configure a recipient that will allow us to go 'check the weather':

1) Right click Recipients, select 'Add'.

2) Specify the following:

Full Name: WeatherCheck

recipient: weathercheck

Carrier: NWS-GetData

ID/PIN: Leave blank

Max Chars: 16000

3) Click Apply.

Now that we have the ability to check the weather, we'll need to configure GetAscii to use PageGate's NWS Filter, which can be downloaded here:

<https://www.notepage.net/support/nwsfilter.zip>

You can also find a copy of the script [here in the documentation](#).

Unzip the file in the root \weather\ folder. For example, if you're using the default paths, you would unzip nwsfilter.zip in to c:\PageGateData\weather\

Next, we'll need to configure the filter for your system:

1) Open nwsfilter.xml in your preferred text editor (NotePad, NotePad++, EditPad, etc)

2) Find all lines that say 'INSERT THE PATH TO YOUR WEATHER FOLDER'. Please replace that text with the path to your root folder. For example, you'll see this line in the filter:

```
INSERT THE PATH TO YOUR WEATHER FOLDER\posts\%InputFileNameBase%-  
FilterIndex%.asc
```

If you're using the default paths, you would use:

```
c:\PageGateData\weather\posts\%InputFileNameBase%-FilterIndex%.asc
```

3) Search for YourRecipientOrGroup and replace that with the name of a recipient or group in PageGate. This is the recipient or group that will receive the weather alerts.

For example, if you want the weather alerts to be sent by email, text, page and API call at the same time, you would definitely want to reference a group name.

4) Save and close the filter.

Now that we have the filter primed and ready, we'll need to tell GetAscii to use it and then configure GetAscii to start reading the weather data.

1) Back in PG Admin, go to Interfaces - GetAscii - Template

2) Specify the following in the 'Settings' section of the template:

RawInterfaceFilter = The full path to your nwsfilter.xml file and declare \ as \092*

UseFilterDll = True

*For Example: c:\092PageGateData\092weather\092nwsfilter.xml

3) Click Apply.

4) Go to Interfaces - GetAscii - Settings - Advanced Polling - Files

5) Click Add, then specify the following:

Recipient: *

From: Leave Blank

File(s): *.txt

Path: The path to your \weather\ folder.

If you're using the default paths, you would specify c:\PageGateData\weather\

File Type: Standard

6) Click Apply.

7) Click Add, then specify the following:

Recipient: *

From: Leave Blank

File(s): *.asc

Path: The path to your \weather\posts\ folder.

If you're using the default paths, you would specify c:\PageGateData\weather\posts

File Type: Standard

8) Click Apply.

9) Click Apply again.

The final step is to configure a scheduled message to 'check the weather' every few minutes:

1) Open the PageGate Admin.

2) Right click on the 'WeatherCheck' recipient and select 'Send Message'

- 3) Leave 'From' and 'Message' blank.
- 4) Check 'Repeat message every X minutes'
- 5) Specify the interval at which PageGate should poll the NWS data feed. Fractions of a minute will be expressed as seconds. For example, a value of 0.5 will poll every 30 seconds.
- 6) Click Send.

From that point forward, PageGate will automatically check the weather every X minutes and will deliver alerts, warnings and watches to the specified recipient or group!

Accepting HTTP Requests: Webdata API

PageGate has an HTTP data receiver (webdata.exe) that can receive input from nearly any API or protocol that offers an HTTP/S request like HTTP GET, HTTP POST and WCTP.

This allows you to receive JSON, XML, HTML, WCTP and similar data submissions via HTTP/S request. Received data doesn't need to conform to a single standard as it is fully processed with the [GetASCII](#) interface and, optionally depending on what you want to do with the received data, [Filter Pack](#).

If you want to receive multiple data formats, the [Filter Pack](#) isn't optional and you will need to implement a filtering system to tell PageGate what to do with each format received.

The API itself is a CGI executable named 'webdata.exe' that must be hosted on a web server your sending systems can access.

In some cases, this lets you run IIS on the PageGate server for an intranet accessible API.

If you have something that can submit webforms or post information via HTTP/S, like ticket tracking, records management, monitoring or other types of systems, this provides a way for them to trigger messages in PageGate.

In other cases, you'll need to host the API's executable on your outward facing web server to provide an internet accessible URL, then provide a UNC share path to the directory on the web server where webdata.exe is hosted for PageGate to access.

If you're dealing with an internet or cloud based system, this provides a way for that system to provide information to your PageGate server.

In more practical terms, the file webdata.exe is located in the PageGate program directory and MUST be copied to the webserver hosting the API.

If PageGate is not running on the web server hosting webdata.exe, you'll also need to share the directory where webdata.exe is hosted on the web server and provide the UNC filepath to PageGate's GetASCII interface.

The Webdata API is only a receiver. All data received is ultimately processed by [GetAscii](#) and potentially with assistance from the [filter pack](#).

PageGate's [Filter Pack](#) opens up quite a few possibilities by allowing you to receive multiple data types from a single or multiple sources but it also allows you to create conditional logic for message processing.

As an example, you could configure a filter that changes who should receive an alert or message based on keywords or keyphrases in the received data. For instance, you could scan for the word lunch in the body of the message and direct it to the cafeteria group instead of a general distribution group.

As another example, it is possible receive the WCTP protocol with this API and you could configure a filter that directs all replies from specific senders to a specified destination, like an email distribution list, an SMS to administrators or any number of things.

You could configure a system where messages from a certain phone number or sender ID will always be copied to certain recipients or groups.

One of the other benefits of the filter is that it can do everything described at the same time. You aren't limited in the type, style or number of filters you can apply to your server.

Step 1: Host webdata.exe on your web server

The first thing you'll need to do is host webdata.exe on your web server to provide the return URL for your API or WCTP host. The following steps are for IIS but this can be done with any web server suite.

- 1) Copy webdata.exe from the PageGate program directory to your server's scripting directory.
- 2) Open the IIS Admin and select the server object.
- 3) Open Handler Mappings.
- 4) Right click CGI-exe, left click Edit Feature Permissions
- 5) Check Execute.
- 6) Click OK.

How PageGate Accepts Messages

2

- 7) Select the Server object again.
- 8) Open ISAPI and CGI Restrictions.
- 9) Right click in the box and select Add.
- 10) Specify the following and click OK:
ISAPI or CGI path: Specify the location of webdata.exe

Description: PageGate API Handler

Check 'Allow extension path to execute'

Step 1a: Share the folder webdata.exe is hosted in

If PageGate is not running on the server hosting webdata.exe, you will need to share the directory where webdata.exe is hosted to allow the GetASCII interface to parse data sent to the receiver.

Step 2: Configure GetASCII to process the received data

Each API and protocol will have a unique structure to the data returned to your server and you'll need to tell PageGate how to interpret the data and what to do with it in general.

To do that, you'll want to configure an 'Advanced Polling' rule:

- 1) Open PG Admin.
- 2) Expand Interfaces - GetAscii - Settings - Advanced Polling
- 3) Click Files.
- 4) Click Add.
- 5) Specify the following and click Apply:

Recipient: *

From: Leave Blank

File(s): *.asc

Path: The path to the folder webdata.exe is stored in on the web server.

Example: \\WebServer01\pgscripts\

File Type: Custom

How PageGate Accepts Messages

2

NOTE: If you will be implementing a [filtering system](#), you will do so at this point. For more information, please see [PageGate's Filter Script documentation](#).

6) Go to Interfaces - GetAscii - Settings - Advanced Polling - Record

7) Specify the following and click Apply:

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

8) Go to Interfaces - GetAscii - Settings - Advanced Polling - Recipient. This section tells PageGate where in the data to look for the recipient or group this reply should go to.

9) Select 'Dynamic Recipient' and configure the parameters.

For example, let's say you're working with the WCTP protocol and want to auto-respond to the person who sent the message. In WCTP, you can key off of senderID="Value".

As another example, let's say you're receiving an XML form and one of the tags will determine who needs to receive the reply. You would key off of <XML Tag>Value</XML Tag>.

Start Pattern: This value determines the first character in the recipient block.

For example, if you're working with WCTP and want to key off of senderID="Value". You would use senderID=" as the Start Pattern.

As another example, if you're working with XML and want to key off of <XML Tag>Value</XML Tag>. You would use <XML Tag> as the Start Pattern.

Start Offset: The number of characters in the start pattern.

End Pattern: This value determines the last character in the recipient block.

For example, if you're working with WCTP and are using senderID=" as the Start Pattern, you would use " as the End Pattern.

As another example, if you're working with XML are using <XML Tag> as the Start Pattern, you would use </XML Tag> as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern.
Examples: -1, -10, etc

How PageGate Accepts Messages

2

10) Go to Interfaces - GetAscii - Settings - Advanced Polling - Sender. This section tells PageGate where in the data to look for the sender value.

11) For most implementations, you can select 'Static Sender' and specify a sender value. However, if the sender's name will be contained in the data read, you can select 'Dynamic Sender' and parameters are configured similarly to the recipient section.

Start Pattern: This value determines the first character in the sender block.

For example, if you're working with WCTP and want to use the value in `senderID="Value"`. You would use `senderID="` as the Start Pattern.

As another example, if you're working with XML and want to key off of `<XML Tag>Value</XML Tag>`. You would use `<XML Tag>` as the Start Pattern.

Start Offset: The number of characters in the start pattern.

End Pattern: This value determines the last character in the sender block.

For example, if you're working with WCTP and are using `senderID="` as the Start Pattern, you would use `"` as the End Pattern.

As another example, if you're working with XML are using `<XML Tag>` as the Start Pattern, you would use `</XML Tag>` as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern.

Examples: -1, -10, etc

12) Go to Interfaces - GetAscii - Settings - Advanced Polling - Message. This section tells PageGate what in the data should be considered the message and also allows for basic reformatting of messages.

13) Select 'Dynamic Message' and configure the parameters. If you want everything in the received data to be considered the message, you would use a Start Pattern of `*` and an End Pattern of `~`.

Start Pattern: This value determines the first character in the message block.

For example, if you're working with WCTP and want to use the value in the `<wctp-Alphanumeric>` tag, you would use `<wctp-Alphanumeric>` as the Start Pattern.

As another example, if you're working with XML and want to key off of `<XML Tag>Value</XML Tag>`. You would use `<XML Tag>` as the Start Pattern.

Start Offset: The number of characters in the start pattern.

How PageGate Accepts Messages

End Pattern: This value determines the last character in the sender block.

For example, if you're working with WCTP and are using <wctp-Alphanumeric> as the Start Pattern, you would use </wctp-Alphanumeric> as the End Pattern.

As another example, if you're working with XML are using <XML Tag> as the Start Pattern, you would use </XML Tag> as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern.

Examples: -1, -10, etc

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed.

To enable this feature, check Use Lookup Table and click Edit Table.

For more information, please see the [lookup table](#) section of the documentation.
Controlling Webdata's HTTP Responses

The HTTP response provided by the Webdata API can be controlled and customized by creating a file named 'response.txt' in the same directory on the web server where 'webdata.exe' resides.

If 'response.txt' does not exist, Webdata will provide a default response of:

```
content-type: text/html
status: 200
<html>
<body>
%Filename%
</body>
</html>
```

However, if your sending system requires a specific response or additional information, you can provide its template in the 'response.txt' file.

For example, you can use the Webdata API to host a very basic WCTP receiver by using this as the response template:

```
<?xml version="1.0"?>
<!DOCTYPE wctp-Operation
SYSTEM "http://dtd.wctp.org/wctp-dtd-v1r1.dtd">
<wctp-Operation wctpVersion="wctp-dtd-v1r1" wctpToken="11AA">
  <wctp-SubmitClientResponse>
    <wctp-ClientSuccess
      successCode="200"
      successText="Message accepted for processing"
      trackingNumber="%FileIndex%"
    >
```

How PageGate Accepts Messages

2

```
    Message accepted for processing
  </wctp-ClientSuccess>
</wctp-SubmitClientResponse>
</wctp-Operation>
```

There are two variables that can be used in the Webdata API's response template.

Value	Function
%Filename%	Outputs the name of the file created by submitting data to webdata.exe
%FileIndex%	Outputs the number of the relative-index kept by Webdata. This number resets as data is processed and provides a somewhat unique value for responses.

Example: Receiving Replies via EMAG WCTP

PageGate can be easily integrated with [Verizon's Enterprise Messaging platform \(EMAG\)](#), which supports an enterprise WCTP gateway that supports one-way and two-way messaging.

With one-way implementations, PageGate delivers SMS through the [Verizon EMAG WCTP gateway](#) but no reply is expected and none is monitored for.

With two-way implementations, PageGate delivers SMS through the same gateway but replies are expected. In order to provide those replies to you, the EMAG WCTP API requires a reply address; somewhere to route the replies. PageGate's Webdata API provides the ability to receive those replies and this API must be hosted on an outward facing web server as the reply URL provided to the EMAG WCTP API must be internet accessible.

You can then configure reply processing in a few ways:

- [Static Recipient or Group](#)

This is the most basic form of reply processing. All replies will be delivered to the specified recipient or group.

- [Reply processing with GetASCII API Advanced Polling rules](#)

If you own the GetASCII API, you can feed the reply information through that system for advanced processing. This allows you to modify message content and who the replies should go to but does not allow for conditional modification or conditional sending. For example, you couldn't have the program scan the body of the message for a keyphrase, then completely change who the message is intended to go to based on it. That would require the [PageGate Filter Pack](#).

- [Reply processing with the GetASCII API and the PageGate Filter Pack](#)

Reply messages can be fed through PageGate's [Filter Pack](#) to allow for conditional sending. For instance, with the PageGate's filter pack, you can tell PageGate to conditionally change who receives the message based on what the text in the reply is. As an example, you could configure a filter system so that any time the word lunch shows up in a reply message, that reply is sent to the cafeteria group instead of the general replies group.

You could configure a system where replies from a certain phone number will always be copied to certain recipients or groups.

You could configure a system where certain keywords or phrases are replaced with other keywords or phrases. For example, you could translate a code like PD4N in to 4th Precinct.

You could configure a system to describe every element I've just described all at once.

However, you must own and implement the [Filter Pack](#) to have the functionality necessary to do conditional sending as described.

Step 1: Host webdata.exe on your web server

The first thing you'll need to do is host webdata.exe on your web server so you can provide the return URL to the EMAG WCTP API. The following steps are for IIS but this can be done with any web server suite.

- 1) Copy webdata.exe from the PageGate program directory to your server's scripting directory.
- 2) Open the IIS Admin and select the server object.
- 3) Open Handler Mappings.
- 4) Right click CGI-exe, left click Edit Feature Permissions
- 5) Check Execute.
- 6) Click OK.
- 7) Select the Server object again.
- 8) Open ISAPI and CGI Restrictions.
- 9) Right click in the box and select Add.
- 10) Specify the following and click OK:

How PageGate Accepts Messages

ISAPI or CGI path: Specify the location of webdata.exe

Description: PageGate API Handler

Check 'Allow extension path to execute'

Step 1a: Share the folder webdata.exe is hosted in

If PageGate is not running on the server hosting webdata.exe, you will need to share the directory where webdata.exe is hosted to allow the GetASCII interface to parse data sent to the receiver.

Step 2: Configure the EMAG WCTP API to send replies to Webdata

- 1) Log in to the [EMAG web portal](#)
- 2) Go to Connections - APIs
- 3) Edit your existing WCTP API connection.
- 4) Provide the URL to webdata.exe in the Return URL field.
Example: https://yourIPhere/pgscripts/webdata.exe
- 5) Apply the changes.

Static Recipient or Group

These are the steps on how to configure PageGate to deliver EMAG WCTP replies to a static recipient or group.

Step 1: Configure GetASCII to process the received data

WCTP has a unique structure to the data returned to your server and you'll need to tell PageGate how to interpret the data and what to do with it in general.

To do that, you'll want to configure an 'Advanced Polling' rule for PageGate's GetASCII interface:

- 1) Open PG Admin.
- 2) Expand Interfaces - GetAscii - Settings - Advanced Polling
- 3) Click Files.
- 4) Click Add.
- 5) Specify the following and click Apply:

Recipient: *

From: Leave Blank

File(s): *.asc

Path: The path to the folder webdata.exe is stored in on the web server.

Example: \\WebServer01\pgscripts\ or c:\inetpub\pgscripts\

File Type: Custom

NOTE: If you will be implementing a [filter](#), you will do so at this point. For more information, please see [PageGate's Filter Script documentation](#).

6) Click Apply, then Apply again.

7) Go to Interfaces - GetAscii - Settings - Advanced Polling - Record

8) Specify the following and click Apply:

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

9) Go to Interfaces - GetAscii - Settings - Advanced Polling - Recipient.

10) Select 'Static Recipient and specify the recipient or group the replies should go to.

11) Go to Interfaces - GetAscii - Settings - Advanced Polling - Sender. This section tells PageGate where in the data to look for the sender value.

12) Select 'Static Sender' and specify a sender value.

13) Go to Interfaces - GetAscii - Settings - Advanced Polling - Message. This section tells PageGate what in the data should be considered the message and also allows for basic reformatting of messages.

14) Select 'Dynamic Message' and configure the parameters. If you want everything in the received data to be considered the message, you would use a Start Pattern of * and an End Pattern of ~.

Start Pattern: <wctp-Alphanumeric>

Start Offset: 19

End Pattern: </wctp-Alphanumeric>

End Offset: -20

How PageGate Accepts Messages

2

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed.

To enable this feature, check Use Lookup Table and click Edit Table.

For more information, please see the [lookup table](#) section of the documentation.

Processing replies with GetAscii Advanced Polling

This is an example of how EMAG WCTP reply messages can be processed with the GetASCII API.

Step 1: Configure GetASCII to process the received data

- 1) Open PG Admin.
- 2) Expand Interfaces - GetAscii - Settings - Advanced Polling
- 3) Click Files.
- 4) Click Add.
- 5) Specify the following and click Apply:

Recipient: *

From: Leave Blank

File(s): *.asc

Path: The path to the folder webdata.exe is stored in on the web server.

Example: \\WebServer01\pgscripts\

File Type: Custom

NOTE: If you will be implementing a [filtering system](#), you will do so at this point. For more information, please see [PageGate's Filter Script documentation](#).

- 6) Go to Interfaces - GetAscii - Settings - Advanced Polling - Record
- 7) Specify the following and click Apply:

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

8) Go to Interfaces - GetAscii - Settings - Advanced Polling - Recipient. This section tells PageGate where in the data to look for the recipient or group this reply should go to.

9) Select 'Dynamic Recipient' and configure the parameters.

Start Pattern: This value determines the first character in the recipient block.

Since we're working with WCTP, you could key off of senderID="Value" to auto-respond to the person that sent the message.

You would use senderID=" as the Start Pattern.

Start Offset: The number of characters in the start pattern: 10

End Pattern: This value determines the last character in the recipient block.

Since we're working with WCTP and are using senderID=" as the Start Pattern, you would use " as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern: -1

10) Go to Interfaces - GetAscii - Settings - Advanced Polling - Sender. This section tells PageGate where in the data to look for the sender value.

11) Select 'Static Sender' and specify a sender value.

12) Go to Interfaces - GetAscii - Settings - Advanced Polling - Message. This section tells PageGate what in the data should be considered the message and also allows for basic reformatting of messages.

13) Select 'Dynamic Message' and configure the parameters. If you want everything in the received data to be considered the message, you would use a Start Pattern of * and an End Pattern of ~ with both Offsets set to 0.

Start Pattern: <wctp-Alphanumeric>

Start Offset: 19

End Pattern: </wctp-Alphanumeric>

How PageGate Accepts Messages

2

End Offset: -20

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed.

To enable this feature, check Use Lookup Table and click Edit Table.

For more information, please see the [lookup table](#) section of the documentation.

Processing replies with a Filter Pack

Note: PageGate's [filter pack](#) supports [RegEx expressions](#), which can allow you to implement an incredibly intricate system.

To say that there are many things you can do with a properly coded [filter pack](#) would be an understatement. The [Filter Pack](#) allows you to modify and conditionally modify any part of what it interacts with. It can reassign who a message is delivered to based on words or phrases in a reply, it can modify the body of the message, it can modify the subject line, it can insert pre-determined text and many other things.

For instance, with the filter pack, you can tell PageGate to conditionally change who receives the message based on what the text in the reply is. As an example, you could configure a filter system so that any time the word lunch shows up in a reply message, that reply is sent to the cafeteria group instead of the general replies group.

You could configure a system where replies from a certain phone number will always be copied to certain recipients or groups.

You could configure a system where certain keywords or phrases are replaced with other keywords or phrases. For example, you could translate a code like PD4N in to 4th Precinct.

You could configure a system to describe every element I've just described all at once, it's just a matter of coding your [filter pack](#) to perform the actions you want.

Step 1: Configure GetASCII to process the received data

WCTP has a unique structure to the data returned to your server and you'll need to tell PageGate how to interpret the data and what to do with it in general.

To do that, you'll want to configure an 'Advanced Polling' rule for PageGate's GetASCII interface:

- 1) Open PG Admin.
- 2) Expand Interfaces - GetAscii - Settings - Advanced Polling

3) Click Files.

4) Click Add.

5) Specify the following and click Apply:

Recipient: *

From: Leave Blank

File(s): *.asc

Path: The path to the folder webdata.exe is stored in on the web server.

Example: \\WebServer01\pgscripts\ or c:

\\inetpub\pgscripts\

File Type: Custom

6) Click Apply, then Apply again.

NOTE: Since you will be implementing a [filtering system](#), you will do so at this point. For more information, please see [PageGate's Filter Script documentation](#).

API Polling

With the [HTTPGET](#) and [HTTPPOST](#) protocols, you can poll information from any Web or REST API that supports the GET or POST methods. That information can then be fed through a [filter pack](#) to turn it in to a message to your recipients and groups.

For instance, if you have a web resource that provides XML content, you can configure PageGate to poll for that XML content, then feed it through a filter pack to turn it in to a readable text message.

As an example, you can poll the [Spillman CAD API](#) for information to dispatch for first responders.

As another example, you can [poll the National Weather Service](#) for information on the weather in your local area, then send that weather information out as an SMS to your recipients and groups.

How PageGate Accepts Messages

2

Example: Spillman XML API

Note: This requires the [GetASCII API](#) and the [Filter Pack](#).

In this example, we're using a Powershell script to automatically poll the Spillman XML API for information in real time, then parsing the collected data with PageGate's GetAscii interface and a filtering system.

One key component of this implementation is understanding what you want done with the data you get from CAD. Do you want every notification or call type to go to every district? Should only certain agency codes receive messages? Understanding the answers to questions like these is key to knowing whether you need to implement a [filtering system](#).

Step 1: Create the script to poll the Spillman XML API

1) In Windows, create a new directory for the script to reside in.

Example: C:\PageGateData\sp-pg-ps\

2) Open your preferred text editor (Notepad, Wordpad, EditPad, NotePad++, etc)

3) Copy and paste the [Spillman API Poller](#) in to your new, blank document.

4) Modify \$WorkingDir to reflect the path you created in step 1.

Example: \$WorkingDir = "C:\PageGateData\sp-pg-ps\"

5) Modify \$RequestUri to reflect the IP address and port of your Spillman XML API host.

Example: \$RequestUri = "127.0.0.1:4081/DataExchange/REST"

6) Modify \$APIRequest to reflect the information you want from Spillman. This API

Request will directly determine what information is requested from CAD.

The default request information from the CAD Active Call table and asks for Law, Fire and EMS call types but NOT Miscellaneous call types.

Default Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<PublicSafetyEnvelope version="1.0">
<From/>
<To/>
<Creation/>
<PublicSafety id="">
<Query>
<CADActiveCallTable>
<CallTypeLawFireEMS search_type="not_equal_to">m</CallTypeLawFireEMS>
```

How PageGate Accepts Messages

```
<CADMasterCallCommentsTable childField="LongTermCallID"
parentField="LongTermCallID" />
</CADActiveCallTable>
</Query>
</PublicSafety>
</PublicSafetyEnvelope>
```

7) Modify \$LogFile to reflect the path you created in step 1.

Example: \$LogFile = "c:\PageGateData\sp-pg-ps\log.txt"

8) Save the file as pollspillman.ps1 in the directory you created in step 1.

Example: c:\PageGateData\sp-pg-ps\pollspillman.ps1

Step 2: Create an Eveng Log source for the script

1) Run Powershell as administrator.

2) Type the following and hit enter:

```
New-EventLog -LogName Application -Source "SpillmanXmlApiRequester"
```

3) Close Powershell.

Step 3: Create a Windows Service for the script

1) There are a few service manager utilities you could use but we recommend using [NSSM](https://nssm.cc/download) (<https://nssm.cc/download>). Unzip the nssm-verzion.zip file in to your script's directory.

Example: c:\PageGateData\sp-pg-ps\nssm-2.24\

2) Open your preferred text editor (Notepad, Wordpad, EditPad, NotePad++, etc)

3) Copy and paste the [Spillman to PageGate Service Installation Script](#) in to your new, blank document.

4) Modify \$NssmPath to reflect the directory you unzipped NSSM to.

Example: \$NssmPath = "c:\PageGateData\sp-pg-ps\nssm-2.24\win64\"

5) Modify \$PsPath to point to the pollspillman.ps1 script you created above.

Example: \$PsPath = "c:\PageGateData\sp-pg-ps\pollspillman.ps1"

6) Save the file as install.ps1 in the directory the script resides in.

Example: c:\PageGateData\sp-pg-ps\

7) Run Powershell as administrator.

8) Change directory in to the folder the script resides in.

How PageGate Accepts Messages

2

Example: c:\PageGateData\sp-pg-ps\

9) Type the following and hit enter:

```
.install.ps1
```

10) Close Powershell.

11) Open the Services list in Windows.

12) Go in to the properties of the Spillman-PageGate-PowerScript service.

13) Select the 'Log On' tab

14) Specify a service account with permission to run powershell scripts. A normal user account is usually sufficient but this user must have logged in to Windows and run Microsoft Edge or the Internet Explorer at least once.

15) Start the Spillman-PageGate-PowerScript service.

NOTE: To remove the script, use the [Spillman to PageGate service removal script](#).

Step 4: Configure GetAscii

This part of the configuration will be determined by what you want to do with the data retrieved from Spillman's API.

- **Basic Implementation**

In the most basic configuration, PageGate will match the value in <AgencyCode> to its recipients and groups list, allowing any call to that agency to go through regardless of incident type or nature code used.

- **Implementation with filtering**

With PageGate's filter pack, you have the ability to determine what dispatches should trigger outbound messages by keying off of any data returned by the API.

Examples of what you can do with filtering:

- Specify which notification types an agency should receive
- Determine whether additional assets should be messaged based on address and notification type
- Restrict certain notification types to only go to specific agencies or assets

How PageGate Accepts Messages

2

Spillman API Poller

Please contact NotePage Tech Support at 781-829-0500 x 2 or support@notepage.com for access to the Spillman API poller and its' related files

Implementing GetAscii for Spillman

Without filtering, PageGate will monitor all received information from the API and dispatch any message addressed to an agency code that exists as a recipient or group in PageGate.

1) In Windows, create a directory for GetAscii to reference.

Example: c:\PageGateData\ASCII\

2) Run PG Admin.

3) Go to Interfaces - GetAscii - Settings

4) Set 'Polling Directory' to the directory the script resides within. Example: c:\PageGatedata\ASCII\

5) Check 'Enabled' and click Apply.

6) Select 'Yes'

7) Go to Interfaces - GetAscii - Settings - Advanced Polling - Files

8) Click Add

9) Specify the following and click Apply, then click Apply again.

Recipient: *

From: CAD

File(s): *.asc

Path: The path to your script.

Example: C:\PageGateData\sp-pg-ps\

File Type: Custom

10) Go to Interfaces - GetAscii - Settings - Advanced Polling - Record

11) Specify the following and click Apply

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

12) Go to Interfaces - GetAscii - Settings - Advanced Polling - Recipient

13) Select 'Dynamic Recipient', specify the following and click Apply

Start Pattern: <AgencyCode>

Start Offset: 12

End Pattern: </AgencyCode>

End Offset: -13

14) Go to Interfaces - GetAscii - Settings - Advanced Polling - Sender

15) Select 'Static Sender', enter the name 'CAD' and click Apply

16) Go to Interfaces - GetAscii - Settings - Advanced Polling - Message

17) Specify the following and click Apply

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

Implementing GetAscii and Filtering with Spillman

PageGate's [filtering system](#) allows you to do nearly anything you want with processed traffic. In our [example filter pack for Spillman](#), we provide examples for how to:

- Only allow an agency to be messaged for specific nature codes.
- Allow an agency to receive all nature codes.
- Determine whether additional assets should be messaged based on address and notification type
- Restrict certain notification types to only go to specific agencies or assets

Step 1: Configure GetAscii

1) In Windows, create a directory for GetAscii to reference.

Example: c:\PageGateData\ASCII\

2) Run PG Admin.

3) Go to Interfaces - GetAscii - Settings

- 4) Set 'Polling Directory' to the directory the script resides within. Example: c:\PageGatedata\ASCII
- 5) Check 'Enabled' and click Apply.
- 6) Select 'Yes'
- 7) Go to Interfaces - GetAscii - Settings - Advanced Polling - Files
- 8) Click Add
- 9) Specify the following and click Apply, then click Apply again.

Recipient: *

From: CAD

File(s): *.asc

Path: The path to your script.

Example: C:\PageGateData\sp-pg-ps\

File Type: Standard

Step 2: Configure your filter(s)

PageGate's filtering system allows you to do nearly anything you want with processed traffic. In our [example filter pack for Spillman](#), we provide examples for how to:

- Only allow an agency to be messaged for specific nature codes.
- Allow an agency to receive all nature codes.
- Determine whether additional assets should be messaged based on address and notification type
- Restrict certain notification types to only go to specific agencies or assets

Step 3: Implement your filter

- 1) In PG Admin, go to Interfaces - GetAscii - Template
- 2) Specify the following in the 'Settings' section:
RawInterfaceFilter = The path to and file name of your script.
Example:
RawInterfaceFilter = c:\PageGateData\sp-pg-ps\pgfilter.xml
- 3) Click Apply.

How PageGate Accepts Messages

2

Example Filter Pack for Spillman

```
<DebugFile>
  <Active>
    True
  </Active>
  <Name>
    debug.txt
  </Name>
</DebugFile>

<Filter>
  <Comment>
    Check to see if processed data is original CAD output
  </Comment>
  <Search>
    <Section>
      Recipient
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      fromcad
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      True
    </Text>
  </Replace>
</Filter>

<Filter>
  <Comment>
    abort if not original CAD output
  </Comment>
  <Search>
    <Section>
```

How PageGate Accepts Messages

```

    Flag
  </Section>
  <Type>
    Literal
  </Type>
  <Text>
    False
  </Text>
</Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      True
    </Text>
  </Replace>
<Abort>True</Abort>
</Filter>

<Filter>
  <Comment>
    set recipient to an invalid name so if there aren't any matches, the file won't
    be
    accepted by PageGate
  </Comment>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      RegEx
    </Type>
    <Text>
      (?s).*
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>

```

```
<Scope>
  Section
</Scope>
<Text>
  nobody
</Text>
</Replace>
</Filter>
```

```
<Filter>
```

```
  <Comment>
```

In this example, we want to restrict what notification types will trigger a message to an agency. In this example, if the incident nature is AIRCRAFT EMERGE or AIRCRAFT HIJACK or ARSON or BOMB THREAT or DROWNING or ESCAPE PRISONER or EXPLOSION or FIRE CHEM SPILL or FIRE FUEL SPILL or FIRE WILDLAND or FIRE STRUCTURE or GAS LEAK INSIDE or GAS LEAK OUTSID or GUNSHOT WOUND or MUTUAL AID OUT or PURSUIT or RAILROAD INCIDE or RESCUE or SCHOOL LOCKDOWN or SEARCH or SMOKE INVESTIGA or UNATTENDED DEAT or WEATHER INCIDEN, the filter will dispatch the message to the recipient or group referenced in the 'Text' section in the <Replace> statement.

```
  </Comment>
```

```
  <Search>
```

```
    <Section>
```

```
      Message
```

```
    </Section>
```

```
    <Type>
```

```
      RegEx
```

```
    </Type>
```

```
    <Text>
```

```
(?s).*<IncidentNature>(AIRCRAFT EMERGE|AIRCRAFT HIJACK|ARSON|BOMB THREAT|DROWNING|ESCAPE PRISONER|EXPLOSION|FIRE CHEM SPILL|FIRE FUEL SPILL|FIRE WILDLAND|FIRE STRUCTURE|GAS LEAK INSIDE|GAS LEAK OUTSID|GUNSHOT WOUND|MUTUAL AID OUT|PURSUIT|RAILROAD INCIDE|RESCUE|SCHOOL LOCKDOWN|SEARCH|SMOKE INVESTIGA|UNATTENDED DEAT|WEATHER INCIDEN)</IncidentNature>
```

```
  </Text>
```

```
</Search>
```

```
  <Replace>
```

```
    <Section>
```

```
      Recipient
```

```
    </Section>
```

```
  <Scope>
```

```
        Section
    </Scope>
    <Text>
        ExampleRecipientOrGroup
    </Text>
</Replace>
<WriteFile>
    THE PATH TO YOUR SCRIPT'S FOLDER\%InputFileNameBase%-%
FilterIndex%%InputFileNameExtension%
</WriteFile>
</Filter>

<Filter>
    <Comment>
        In this example, we want to make sure the "Search and Rescue" team is
        ALWAYS messaged in addition to the original agency dispatched any time the call type is
        'SEARCH' is used.
    </Comment>
    <Search>
        <Section>
            Message
        </Section>
        <Type>
            RegEx
        </Type>
        <Text>
            (?s).*<IncidentNature>SEARCH</IncidentNature>

        </Text>
    </Search>
    <Replace>
        <Section>
            Recipient
        </Section>
        <Scope>
            Section
        </Scope>
        <Text>
            YourSearchAndRescueRecipientOrGroup
        </Text>
    </Replace>
</WriteFile>
```

How PageGate Accepts Messages

2

```
THE PATH TO YOUR SCRIPT'S FOLDER\%InputFileNameBase%-%
FilterIndex%%InputFileNameExtension%
  </WriteFile>
</Filter>
```

```
<Filter>
```

```
  <Comment>
```

In this example, we want to make sure the "Search and Rescue" team is ALWAYS messaged in addition to the original agency dispatched any time the call type is 'RESCUE' is used.

```
  </Comment>
```

```
  <Search>
```

```
    <Section>
```

```
      Message
```

```
    </Section>
```

```
    <Type>
```

```
      RegEx
```

```
    </Type>
```

```
    <Text>
```

```
(?s).*<IncidentNature>RESCUE</IncidentNature>
```

```
  </Text>
```

```
</Search>
```

```
  <Replace>
```

```
    <Section>
```

```
      Recipient
```

```
    </Section>
```

```
    <Scope>
```

```
      Section
```

```
    </Scope>
```

```
    <Text>
```

```
      YourSearchAndRescueRecipientOrGroup
```

```
    </Text>
```

```
  </Replace>
```

```
  <WriteFile>
```

```
THE PATH TO YOUR SCRIPT'S FOLDER\%InputFileNameBase%-%
FilterIndex%%InputFileNameExtension%
  </WriteFile>
</Filter>
```

```
<Filter>
```

```
  <Comment>
```

How PageGate Accepts Messages

2

In this example, we're scanning the address for a specific value. If it finds that value, the agency referenced should be messaged in addition to the original agency dispatched. This can be used to dispatch additional resources when an event takes place at a school, hospital or other specifically addressed location.

```
</Comment>
<Search>
  <Section>
    Message
  </Section>
  <Type>
    RegEx
  </Type>
  <Text>
    (?s).*<RespondToAddress>An Example Address</RespondToAddress>

    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      ExampleRecipientOrGroup
    </Text>
  </Replace>
  <WriteFile>
    THE PATH TO YOUR SCRIPT'S FOLDER\%InputFileNameBase%-%
    FilterIndex%%InputFileNameExtension%
  </WriteFile>
</Filter>

<Filter>
  <Comment>
    set recipient back to an invalid name in case the above filter changed it
  </Comment>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
```



```
    RegEx
  </Type>
  <Text>
      (?s).*
  </Text>
</Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      nobody
    </Text>
  </Replace>
</Filter>

<Filter>
  <Comment>
    In this example, the specified agency should receive all call types.
  </Comment>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      RegEx
    </Type>
    <Text>
      <AgencyCode>ExampleAgencyCode</AgencyCode>
    </Text>
  </Search>

  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
```

```
        ExampleAgencyCode
    </Text>
</Replace>

<Continue>
    False
</Continue>
</Filter>
```

TCP Server Module

This interface allows PageGate to function as a TCP server to receive TCP packets from a variety of sources. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#).

Step 1: Create a directory for the TCP Server

- 1) In Windows, create a new directory for the TCP Server. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called TCP as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\TCP).
- 2) In Windows, browse in to the PageGate [program directory](#).
- 3) Move the pgtcpserver.exe and pgtcp.ini files to the folder you created in step 1.

Step 2: Configure the TCP Server's settings

- 1) The first thing we'll need to do is modify the pgtcp.ini file. Open the file in notepad.

To give you a little more information, this INI file controls the behavior of the TCP Server. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

- 2) Configure the port the TCP server should listen to. By default, the port is set to 5002.
- 3) Change the OutputFolder= value to the directory you created in step 1. By default, C:\PageGateData\TCP\
- 4) Change the LogFolder= value to your [logging directory](#). If you are unsure of what this folder is, open the PageGate Admin and go to the [Program - Settings](#) section. By default, C:\PageGateData\Logs\

How PageGate Accepts Messages

2

5) Next, we'll need to set the end of record delimiter but let's go in to what that means first. When the TCP Server is receiving information, it doesn't really know what in the data is important; it's just receiving TCP packets of data. The end of record delimiter tells the TCP server that the expressed series of characters denotes a break in the data stream.

For example, the TCP server could receive a series of 5 packets:

- a) This is
- b) a
- c) test message #
- d) This is a new
- e) message

If you set EndOfRecordDelimiter = #, that tells the TCP server that everything before every # received is a data block to be parsed.

Put another way, the end of record delimiter tells the TCP server how to break down the information it's receiving in to processable chunks.

It's the "over" of a CB radio conversation.

5) Below that, there are RequiredText and RequiredTextOperator values that can be set.

RequiredText specifies text that must appear for a data block to be processed and you may specify multiple values by separating each with a | (pipe).

The RequiredTextOperator specifies whether you're using OR or AND logic.

For example, let's say you specify the following:

```
RequiredText=alarm|notification  
RequiredTextOperator=AND
```

That tells the TCP server that both the words alarm AND notification need to appear in a received data block for that data block to be treated as a message to send.

As another example:

```
RequiredText=alarm|alert  
RequiredTextOperator=OR
```

That tells the TCP server that either the words alarm or alert need to appear in a received data block for that data block to be treated as a message to send.

6) Next we have EndOfRecordTimeout, whose value is expressed in milliseconds.

If data is received by the TCP server, but the EndOfRecordDelimiter isn't received within the time specified, the data will be accepted for processing anyway.

By default, this is set to 3000 milliseconds or 3 seconds.

7) `RejectOnTimeout` determines whether timeouts should post a rejection notice or not. By default, this is set to true.

8) `AcceptResponse` determines what the TCP server will echo to the sending system upon acceptance of a data block.

9) `RejectResponse` determines what the TCP server will echo to the sending system upon rejection of a data block.

10) Last, we have two optional settings: `StaticRecipient` and `StaticSender`

`StaticRecipient` allows you to specify the name of a specific recipient or group in PageGate that ALL traffic from the TCP server should be passed through.

`StaticSender` allows you to hard code a sender's name that should be associated with ALL traffic from the TCP server.

11) Save the .ini file and close notepad.

Step 3: Run the TCP Server

1) Open a command prompt with elevated permission.

2) Change directory to the TCP Server's program directory. By default, `c:\PageGateData\TCP\`

3) Type the following and hit enter:
`pgtcpserver /install`

4) Exit the command prompt.

5) Open the Services list in Windows.

6) Go in to the Properties of the PageGate TCP Server service.

7) Change the Startup Type to Automatic and Start the Service.

Step 4: Configure the GetASCII API to process the TCP Server's messages

1) Go to Interfaces - GetASCII - Settings.

2) Set the 'Polling Directory' to the output folder. By default, `C:\PageGateData\TCP\`

- 3) Check Enabled.
- 4) Click Apply.
- 5) Go to Program - Settings.
- 6) In the 'Run on this server' section, check GetASCII
- 7) Click Apply.
- 8) If prompted for credentials, leave them blank and click Apply again.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the TCP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc and it is a Standard file type.

UDP Server Module

This interface allows PageGate to function as a UDP server to receive UDP packets from a variety of sources. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#).

Step 1: Create a directory for the UDP Server

- 1) In Windows, create a new directory for the UDP Server. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called UDP as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\UDP\).
- 2) In Windows, browse in to the PageGate [program directory](#).
- 3) Move the pgudpserver.exe and pgudp.ini files to the folder you created in step 1.

Step 2: Configure the UDP Server's settings

- 1) The first thing we'll need to do is modify the pgudp.ini file. Open the file in notepad.

To give you a little more information, this INI file controls the behavior of the UDP Server. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

- 2) Configure the port the UDP server should listen to. By default, the port is set to 5002.

3) Change the OutputFolder= value to the directory you created in step 1. By default, C:\PageGateData\UDP\

4) Change the LogFolder= value to your [logging directory](#). If you are unsure of what this folder is, open the PageGate Admin and go to the [Program - Settings](#) section. By default, C:\PageGateData\Logs\

5) Next, we'll need to set the end of record delimiter but let's go in to what that means first. When the UDP Server is receiving information, it doesn't really know what in the data is important; it's just receiving UDP packets of data. The end of record delimiter tells the UDP server that the expressed series of characters denotes a break in the data stream.

For example, the UDP server could receive a series of 5 packets:

- a) This is
- b) a
- c) test message #
- d) This is a new
- e) message

If you set EndOfRecordDelimiter = #, that tells the UDP server that everything before every # received is a data block to be parsed.

Put another way, the end of record delimiter tells the UDP server how to break down the information it's receiving in to processable chunks.

It's the "over" of a CB radio conversation.

5) Below that, there are RequiredText and RequiredTextOperator values that can be set.

RequiredText specifies text that must appear for a data block to be processed and you may specify multiple values by separating each with a | (pipe).

The RequiredTextOperator specifies whether you're using OR or AND logic.

For example, let's say you specify the following:

```
RequiredText=alarm|notification  
RequiredTextOperator=AND
```

That tells the UDP server that both the words alarm AND notification need to appear in a received data block for that data block to be treated as a message to send.

As another example:

```
RequiredText=alarm|alert  
RequiredTextOperator=OR
```

That tells the UDP server that either the words alarm or alert need to appear in a received data block for that data block to be treated as a message to send.

6) Next we have EndOfRecordTimeout, whose value is expressed in milliseconds. This value tells the UDP server how long to wait before timeout after receiving data, to accept a data record.

By default, this is set to 3000 milliseconds or 3 seconds.

7) RejectOnTimeout determines whether timeouts should post a rejection notice or not. By default, this is set to true.

8) AcceptResponse determines what the UDP server will echo to the sending system upon acceptance of a data block.

9) RejectResponse determines what the UDP server will echo to the sending system upon rejection of a data block.

10) Last, we have two optional settings: StaticRecipient and StaticSender

StaticRecipient allows you to specify the name of a specific recipient or group in PageGate that ALL traffic from the UDP server should be passed through.

StaticSender allows you to hard code a sender's name that should be associated with ALL traffic from the UDP server.

11) Save the .ini file and close notepad.

Step 3: Run the UDP Server

1) Open a command prompt with elevated permission.

2) Change directory to the UDP Server's program directory. By default, c:\PageGateData\UDP\

3) Type the following and hit enter:
pgudpserver /install

4) Exit the command prompt.

5) Open the Services list in Windows.

6) Go in to the Properties of the PageGate UDP Server service.

7) Change the Startup Type to Automatic and Start the Service.

Step 4: Configure the GetASCII API to process the UDP Server's messages

- 1) Go to Interfaces - GetASCII - Settings.
- 2) Set the 'Polling Directory' to the output folder. By default, C:\PageGateData\UDP\
- 3) Check Enabled.
- 4) Click Apply.
- 5) Go to Program - Settings.
- 6) In the 'Run on this server' section, check GetASCII
- 7) Click Apply.
- 8) If prompted for credentials, leave them blank and click Apply again.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the UDP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc and it is a Standard file type.

GetAscii Integration

[The GetAscii API](#), also known as the Command Line/ASCII interface, is capable of monitoring a directory or series of directories on the local hard drive or across the network for ascii file output and also provides a command line interface for other applications to use. This module allows many 'off the shelf' and custom applications, such as SolarWinds' Network Performance Monitor, SolarWinds' Orion, Paessler's PRTG and several CAD programs, among many others, to tell PageGate when a message needs to be sent.

[GetAscii](#) can accept messages from other applications in the following ways:

- [Command Line](#)
- [ASCII Text/XML/HTML/JSON File](#)
- [Registry Polling](#)

[Click here](#) for more information at the [GetAscii](#) section of the documentation.

You may also want to review the sections on:

- [How to see if the GetAscii module is running](#)

How PageGate Accepts Messages

2

- [How to get a list of current PageGate Recipients and Groups](#)

Command Line Integration

PageGate includes two executable files that can be run to pass messages into its message queue: `sendpage.exe` and `sendpage32.exe`.

`Sendpage.exe` is a 16-bit program, and `sendpage32` is a 32-bit program. Depending on how the programs are to be called, you may need to use one or the other.

To use the executables, run them and provide the recipient, sender, and message information as parameters on the command line.

Example:

```
sendpage32 Joe Brian Testing 123
```

That example would send a message to Joe from Brian with the message Testing 123.

These executables can be hosted in any directory, whether it is local to the PageGate Server or a network directory. If you need to call the command line executables from another location, please reference that path before the file name. For example, you would use this as an example command to a network instance of the command line executable:

```
\\servername\directory\sendpage32.exe Joe Brian Testing 123
```

or you would use this command to a local instance of the command line executable:

```
c:\PageGateData\ASCII\sendpage32.exe Joe Brian Testing 123
```

Any time a command is issued to a command line executable, the CLE outputs a file in the same directory that the `GetAscii` module will read in for data parsing.

Detailed information about the use of the `sendpage` programs can be found in the [GetAscii](#) section (under [Interfaces](#)) of the documentation.

Text File Integration

[GetAscii](#) can extract message information from plaintext formatted ascii files, whether the content in the file is HTML, XML, JSON or other form of plaintext.

[GetAscii's](#) basic polling module looks for text files with a predefined file format in a single location. [GetAscii](#) also supports an [advanced polling module](#), which enable [GetAscii](#) to read files from multiple folders and from files that aren't in a standard format. [GetAscii](#) can even 'watch' files for additions or changes, then extract message data from them.

Detailed information about the use of the [GetAscii](#) text file interface can be found in the [GetAscii](#) section (under [Interfaces](#)) of the documentation.

Registry Polling Integration

The GetAscii interface can accept messages that are passed to it via the Windows registry. Messages written to a predefined location in the registry will be picked up by the GetAscii interface. The application to be integrated with PageGate MUST be running on the same computer as the GetAscii module to use this integration method.

To integrate using this method:

- 1) Open the Windows Registry Editor.
- 2) Browse to
HKEY_LOCAL_MACHINE\Software\Wow6432Node\NotePage\Transfer\Messages\
Note: In versions of Windows prior to Windows Vista and Windows Server 2008, the path is HKEY_LOCAL_MACHINE\Software\NotePage\Transfer\Messages\
3) Create a new Key.
- 4) The name of the new key should be unique so it doesn't get overwritten by the next message.

Under the new key, the following values should be created:

Name	Type	Value	Description
Ready	Dword	0	Specifies whether this message is ready to be read in.
To	String	Custom	This entry specifies the Recipient or Group within PageGate who should receive this message.
From	String	Custom	This entry specifies the Sender value for this message.
Message	String	Custom	This entry specifies the text of the message to be sent.

After all the values are written, set the 'Ready' value to one (1). This signals PageGate that the entry is complete and can be read in. These new entries are automatically removed by the [GetAscii](#) interface after they are read in.

How to see if GetAscii is active

Read the PageGate database file (pagegate.mdb)

The current location of the PageGate database can be read from the registry:
HKEY_LOCAL_MACHINE\Software\NotePage\Transfer\DBDir

How PageGate Accepts Messages

2

It is in a format compatible with MS Access 2000. In the 'Control' table, the 'getascii_state' field will be True if the GetAscii module is running.

Read the registry

The registry value HKEY_LOCAL_MACHINE\Software\NotePage\Transfer\Running will have a value of 1 when GetAscii is running

Use the OpenEvent API call

When GetAscii starts, it creates a Windows event named 'NotePagelsRunning'. This event is destroyed when GetAscii stops. The API call OpenEvent(0,0,"NotePagelsRunning") will return a non-zero result if the GetAscii interface is running.

How to get a list of Recipients and Groups

Read the PageGate database file (pagegate.mdb)

The current location of the PageGate database can be read from the registry:
HKEY_LOCAL_MACHINE\Software\NotePage\Transfer\DBDir

It is in a format compatible with MS Access 2000. The 'Users' table has an entry for each valid PageGate recipient and group.

Read the registry

The GetAscii interface keeps an up-to-date list of all the PageGate recipients and groups in the Windows registry:

HKEY_LOCAL_MACHINE\Software\NotePage\Transfer\Recipients

HKEY_LOCAL_MACHINE\Software\NotePage\Transfer\Groups

Filter Pack

The filter pack is a scripting system that allows you to modify any input in to or output from any of PageGate's interfaces based on nearly any criteria of your choosing.

Things you can do with the filtering system:

- Conditionally modify who a message is intended to go to based on key words or key phrases.
 - All fire, rescue and accident calls should be dispatched but non-emergency calls should not be.
 - Only alarms with "CRITICAL" in them should be delivered.
- Exam
ples:
 - Messages with the keyphrase "HAZARD" should go to a high priority group instead of the normal distribution list.
 - Replies with the word 'subscribe' should perform one action but replies with the word 'unsubscribe' should perform another.

- Modify the contents or structure of data to turn it in to something more user-friendly and readable.

- Pull content from an API (like the National Weather Service, a Ticket Tracking system like Jira or ServiceNow, etc)

Exam

- ples:
- Parse XML, HTML, JSON or any other form of delimited text in to readable message content

- Trigger a Powershell or other script.

Descr
iption:

- This allows you to have PageGate trigger other applications or processes by tying them to a conditional filter in PageGate. Powershell is incredibly powerful and with the right script, you could have PageGate execute functions that would modify another application's database, perform network or domain lookup functions or anything else made available by Powershell.

This is certainly not a comprehensive list of things you could do and, effectively, this filter allows you to modify any input fed in to or output delivered from PageGate based on nearly any selection criteria of your choosing.

Over the course of this documentation, we'll be going over how to configure, code and implement PageGate's filtering system.

First, the filter scripts require the file 'pgfilter.exe' to reside in the PageGate program directory and will need to reference an XML file containing the script, which can reside in any directory the PageGate server has access to (whether that's on the local hard drive or on a network share via UNC path). You should receive all necessary files for the filtering system upon purchase but if you do not have them, please contact our sales and registration department at 781-829-0500 x 1 or by email at sales@notepage.com.

Second, it's important to understand the chain of events that [triggers the script to run](#).

For more detailed information, please see: [How to Implement the Script](#)

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.

How PageGate Accepts Messages

2

?	Single character wildcard.
*	Multiple character wildcard.
	<u>IMPORTANT</u>
	ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.
\	This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ASCII value. The ASCII value of ~ is: \126

Note: PageGate's [filter script](#) supports [RegEx expressions](#), which can allow you to implement an incredibly intricate system.

How to Implement the Script

Step 1: Copy the filter's executable in to the PageGate program directory

The file 'pgfilter.exe' must reside in the PageGate program folder. If you do not have a copy of this file, please contact our sales and registration department at sales@notepage.com or by phone at 781-829-0500.

Step 2: Determine what you want filtered

Filter scripts are called through PageGate's template system and can be broken down in to two categories: Inbound and Outbound

Inbound filters can trigger on and modify content passed to the program. These filters are applied to the program's input interfaces and on a global level to effect all input interfaces.

Outbound filters can trigger on and modify messages delivered by the program. These filters can be applied to a recipient, carrier, connector and on a global level.

You can have a system reference multiple filters for different purposes at the same time.

For example, you could configure an inbound filter to strip XML and JSON formatting from received data as well as a global outbound filter that inserted a custom header before every message. You could then make exceptions to that global rule by configuring unique filters for a carrier, recipient or connector.

You could also configure an SMS reply processing system that filtered reply messages containing certain keywords to specific recipients and groups while also implementing an outbound filter to customize the contents or formatting of messages delivered to them.

You could do everything described all at once. It really is just a matter of determining what you want to do, then implementing an appropriate solution.

Step 3: Write the filter

How PageGate Accepts Messages

2

Each filter exists as an XML file in a directory of your choosing; this directory just needs to be accessible to the PageGate server, whether that's on the local hard drive or via UNC to a network path. The XML file containing your filter may also be called anything you like, you can give the file any name as long as its extension is .xml

[Steps for writing filters can be found here.](#)

Step 4: Reference the filter

After writing a filter, you'll need to add reference it in the 'Settings' section of a template in PG Admin.

Inbound/Interface Filters

Value	Description	Example
InterfaceFilter	<p>This applies a filter to processed data. This allows the interfaces to apply the initial layer of data sorting and collection and then applies the filter to the result.</p> <p><u>GetAscii</u> Standard polling: Filter is applied against temp file with recipient, sender and message Advanced polling: Filter is applied against temp file with recipient, sender and message, and after lookup substitution</p> <p><u>GetMail</u> Filter is applied against temp file with recipient, sender and message, after they have been extracted from the email, and decoding has taken place</p> <p><u>GetSerial</u> Filter is applied against temp file with recipient, sender and message, and after lookup substitution</p> <p><u>GetTap</u> Filter is applied against each message that is received. The file processed by PGFilter will have Id as first line, second line blank, and message on remaining lines.</p> <p><u>GetWeb</u></p>	InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml

	<p>Filter is applied against temp file with recipient, sender and message</p> <p>This can be used on an interface or global level. If used in an interface, it applies the filter to that interface. If used on a global level, it applies the filter to all input from all interfaces.</p>	
<p>RawInterfaceFilter</p>	<p>This applies a filter to raw, unprocessed data. This applies the filter before the interface has a chance to read the data. The specific behavior varies depending on the interface in question.</p> <p><u>GetAscii</u> Standard polling: Filter is applied against the raw text file before GetAscii tries to process it</p> <p>Advanced polling: Filter is applied against each record that is found, before it is processed any further (record pattern matching done first)</p> <p><u>GetMail</u> Filter is applied against the raw file received from the PageGate mail server process. All headers are intact and no decoding has taken place yet. Envelope values for recipient and sender are not available for modification in the raw mail file (because they are read from the envelope).</p> <p><u>GetSerial</u> Filter is applied against each record found before it is processed any further (record pattern matching is done first)</p> <p><u>GetTap</u> Filter is applied against each message that is received.</p> <p><u>GetWeb</u> Filter is applied against raw text file created by webgate.exe CGI program</p>	<p>RawInterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p>

	<p>This can be used on an interface or global level. If used in an interface, it applies the filter to that interface. If used on a global level, it applies the filter to all input from all interfaces.</p>	
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Outbound Filters

Value	Description	Example
Filter	<p>This applies a filter to outbound messages. This can be applied on a recipient, carrier, connector or global level.</p> <p>If used in a recipient template, all messages delivered to that recipient will be processed through the filter.</p> <p>If used in a carrier template, all messages delivered through that carrier will be processed through the filter.</p> <p>If used in the program template, it applies to all output from the program.</p>	<p>ConnectorFilter = c: \PageGateData\Filters\yourfilter.xml</p>
Connector Filter	<p>This applies a filter to the messages delivered by a connector. This filter can only modify the contents of a message and cannot modify the recipient or sender information.</p> <p>This can be applied on a connector or global level.</p> <p>If used in the program template, it applies to all connectors in the program.</p>	<p>Filter = c: \PageGateData\Filters\yourfilter.xml</p>
SubjectFilter	<p>This applies a filter to the subject line of any outbound email. This value is only referenced by the SMTP protocol and is not referenced by other protocols.</p> <p>This can be applied on a recipient, carrier, connector or global level.</p> <p>If used in a recipient template, all SMTP/email messages delivered to that</p>	<p>SubjectFilter = c: \PageGateData\Filters\yourfilter.xml</p>

	<p>recipient will have their subject lines processed through the filter.</p> <p>If used in a recipient template, all SMTP/email messages delivered to through that SMTP/email carrier will have their subject lines processed through the filter.</p> <p>If used in the program template, all SMTP/email traffic sent from the system will have its subject line processed through the filter.</p>	
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Hierarchy

An Interface (and Global) template can have both a RawInterfaceFilter and an InterfaceFilter. Both filters will run in that order if they are both set.

Connectors will run any outbound filter according to which template has the highest priority. In order, they are:

- 1) Recipient
- 2) Carrier
- 3) Connector
- 4) Global

After implementing a filter script, it is highly recommend that you go through a [debugging process](#) to verify what you've configured. See [Debugging the Script](#) for more information. Debugging the Script

To debug the filter script, you'll first need to set the [Debug tag](#) to true.

By setting Debug to True, that puts the script in to debugging mode. When in debugging mode, the script will display a window that shows the result of every component of your filter. It will display whether the filter performed its functions successfully or if there was an error and, if there was an error, what the error was related to. For example, "Match not found" in the event that your search criteria couldn't be found.

This makes tracking down errors in the script much easier than it otherwise would be.

If you're using preproc.bat, rename it to preprocv.bat

If you're using pstproc.bat, rename it to pstprocv.bat

The v in the file name stands for Verbose, so you're effectively putting the script in to a mode that will display far more information on the screen.

The Filter Script

Filters can be simple or complex depending on what you want to do. The most basic filters are search and replace functions, where you remove specific patterns like XML tags from text or replace short-codes with more verbose information. The more advanced filters can conditionally modify who receives a message, what the message says and execute functions via batch file or Powershell script, among other things.

The filters themselves are written in XML and, to start, you may want to look at [the complete filter script template](#). That will give you a general overview of each section and all of the possible values you could use in them.

To break things down, there is one major section, Filter, that determines what each section of the script will do. Additionally, there are two conditional sections, InputFile and OutputFile, plus an optional debugging mode:

- [Filter](#)

The Filter section determines what you want to do with the data read as input. A single filter script may reference multiple individual <Filter> sections to perform multiple tasks within the same data set. It's also important to note that the filter system supports the use of Regex, allowing for a high level of customization and control.

- [InputFile](#)

Most scripts will not need this parameter. When using the filter system in conjunction with the GetAscii interface or a FILE protocol carrier, the InputFile section determines which file(s) you want the filter to scan.

- [OutputFile](#)

Most scripts will not need this parameter. When using the filter system in conjunction with the GetAscii interface or a FILE protocol carrier, the OutputFile section determines what you want to name the files output by the filter's modifications.

- [Debug](#)

The Debug section determines whether you want a debug level review of the filter to appear before the filter completes its actions. This is incredibly useful when first configuring a script and we recommend that this flag be enabled until the filter is ready to put in to production.

For most scripts, you'll want to start with the [Filter](#) section.

The Complete Filter Script Template

Note: The InputFile and OutputFile sections are not required for most scripts. For more information, please see [The Filter Script](#).

<InputFile>

How PageGate Accepts Messages

2

```
<Name>
    (Parameter), static input filename
</Name>
<Delete>
    True, (False) - deletes the Input File, even if <Abort> is used
</Delete>
</InputFile>
<OutputFile> - not used if the .dll version of the filter is running the script
    <Name>
        (InputFile), static output filename
    </Name>
</OutputFile>
<Debug>
    <Console>
        <Active>
            True, (False)
        </Active>
    </Console>
    <File>
        <Active>
            True, (False)
        </Active>
        <Name>
            debug log filename
        </Name>
        <Append>
            True, (False) - appends to existing file instead of recreating it
        </Append>
    </File>
    <LogInputFile>
        True, (False)
    </LogInputFile>
    <LogOutputFile>
        True, (False)
    </LogOutputFile>
</Debug>
<Filter>*
    <Comment>
        optional comment text that will be shown in debug output
    </Comment>
    <FilterName>
        If you are using the <Goto> tag for looping, then the filter you want
        to jump back to must have a name
    </FilterName>
```

<Lookup> - if the Lookup section is included, no other sections are processed (i.e. <MatchFound> and <MatchNotFound> are not run

<Variable>

(All), Recipient, Sender, Message, Temp1-10

</Variable>

<CaseSensitive>

True, (False)

</CaseSensitive>

<SpaceSensitive>

True, (False) - If False, leading and trailing spaces are ignored

(otherwise they need to match exactly)

</SpaceSensitive>

<List>

A list of comma separated values - one per line
or a filename that contains the list

list,

If the text in the <Variable> text matches the first value of a pair in the

the second value in the pair is substituted for it

with | symbols

Multiple values can be included in the first value by separating them

Values can be quoted - if so use two quotes in a row (""") to include
a single quote in the value's text

If no match is found, and the last pair in the list have a * for it's first
value, then this will be used as a default value

character

Comments: any line of text with a pound sign (#) as the very first

(no leading spaces or tabs) will be considered a comment,

and ignored.

</List>

</Lookup>

<Search>

<Type>

(Literal), RegEx, GreaterThan, LessThan - for GreaterThan and
LessThan, <Variable> is GreatherThan/LessThan <Text> </Type>

<Variable>

(All), Recipient, Sender, Message, Temp1-10, Flag

</Variable>

<Text>*

search text, True, False

must be quoted to retain beginning and/or ending whitespace

</Text>

</Search>

<MatchFound> - if you don't include a <Search> section,

<Replace> - multiple Replace sections are allowed

<Type>
(Literal), RegEx, And, Or - And/Or sets Flag equal to Flag

And/Or <Text>

</Type>
<Variable>
(All), Recipient, Sender, Message, Temp1-10, Flag
</Variable>
<Scope>
(Match), Entire - replace only the matched text, or the entire
variable's contents
</Scope>
<ReplaceAll>
True, (False) - replaces all matches in the variable
</ReplaceAll>
<Text>*
replacement text, True, False
must be quoted to retain beginning and/or ending whitespace
</Text>
</Replace>
<WriteFile>
<Name>*
if match found, filename to save to - most macros are valid
</Name>
<Variable>
(All), Recipient, Sender, Message, Temp1-10
</Variable>
</WriteFile>
<Run>
<Command>*
run this command (no spaces in folders or filenames)
</Command>
<Async>
(True), False - if true, the filter continues immediately, if false it
waits for
the program it started to finish
</Async>
<Show> - true give the program a window to run in, false hides the
program
True, (False)
</Show>
</Run>
<RunFilter>
<SettingsFile>*
path and name of new filter settings file

```
</SettingsFile>
<Variable>
    (), All, Recipient, Sender, Message, Temp1-10
    name of the variable that holds the text to be sent to the filter
    * if Variable is used (and not empty), the InputFile value is
ignored
    </Variable>
<InputFile>
    path and name of input file - ignored if Variable section is set
</InputFile>
<DebugFile>
    path and name of debug file
</DebugFile>
<AppendDebug>
    True, (False)
</AppendDebug>
<UseDll>
    True, False - if pgfilter.dll is running, then default value is True,
otherwise it is False
    </UseDll>
</RunFilter>
<ReadFile>
    <Name>*
        if match found, filename to read - most macros are valid
    </Name>
    <Variable>
        (All), Recipient, Sender, Message, Temp1-10
    </Variable>
    <Delete>
        True, (False) - delete the file after reading it
    </Delete>
</ReadFile>
<Goto>
    Name of filter to jump to - *** be careful not to make infinite loops
</Goto>
<Abort>
    True, (False) - exit without writing to the Output File (which is the Input
File by default)
    </Abort>
    <Continue>
        (True), False - whether or not to continue processing the next filter
block
    </Continue>
<MatchFound>
```

How PageGate Accepts Messages

2

<MatchNotFound> - this section is always run if there is no <Search> section

<Replace> - multiple Replace sections are allowed

<Type>
(Literal), And, Or - And/Or sets Flag to Flag And/Or

replacement Text

</Type>

<Variable>
(All), Recipient, Sender, Message, Temp1-10, Flag

</Variable>

<Text>*

replacement text, True, False
must be quoted to retain beginning and/or ending whitespace

</Text>

</Replace>

<WriteFile>

<Name>*

if match found, filename to save to - most macros are valid

</Name>

<Variable>
(All), Recipient, Sender, Message, Temp1-10

</Variable>

</WriteFile>

<Run>

<Command>*

run this command (no spaces in folders or filenames)

</Command>

<Async>
(True), False - if true, the filter continues immediately, if false it

waits for
the program it started to finish

</Async>

<Show> - true give the program a window to run in, false hides the

program
True, (False)

</Show>

</Run>

<RunFilter>

<SettingsFile>*

path and name of new filter settings file

</SettingsFile>

<Variable>
(), All, Recipient, Sender, Message, Temp1-10
name of the variable that holds the text to be sent to the filter

* if Variable is used (and not empty), the InputFile value is

ignored

```
</Variable>
<InputFile>
    path and name of input file - ignored if Variable section is set
</InputFile>
<DebugFile>
    path and name of debug file
</DebugFile>
<AppendDebug>
    True, (False)
</AppendDebug>
<UseDll>
    True, False - if pgfilter.dll is running, then default value is True,
```

otherwise it is False

```
</UseDll>
</RunFilter>
<ReadFile>
    <Name>*
        if match found, filename to read - most macros are valid
    </Name>
    <Variable>
        (All), Recipient, Sender, Message, Temp1-10
    </Variable>
    <Delete>
        True, (False) - delete the file after reading it
    </Delete>
</ReadFile>
<Goto>
    Name of filter to jump back to - be careful not to make infinite loops
</Goto>
<Abort>
    True, (False) - exit without writing to the Output File (which is the Input
```

File by default)

```
</Abort>
<Continue>
    (True), False - whether or not to continue processing the next filter
```

block

```
</Continue>
<MatchNotFound>
</Filter>
<Filter>...
```


How PageGate Accepts Messages

2

Notes:

Sections with * are required

Values in () are default values if section is not present

The first commandline parameter is the name of the InputFile to be processed
If an <InputFile><Name> section is used, its value overrides the commandline parameter value

The default settings file name is pgfilter.xml

The settings file name can be set by using settingsfile= as a parameter on the commandline. The input file name would then be the next parameter

The debug file name can be set by using debugfile= as a parameter on the commandline. The input file name would then be the next parameter

By default, when writing debug info to a file, the file is recreated each time
Using the commandline parameter debugappend=true will append to the file instead

*** WARNING *** don't leave debugappend set for long periods, because the file will keep growing forever

OutputFile, ReadFile, WriteFile support variables (in addition to the normal ones):

- %InputFilePathAndName%
- %InputFilePath%
- %InputFileName%
- %InputFileNameBase%
- %InputFileNameExtension%
- %FilterIndex%

Section values for Recipient, Sender, Message, and All really just mean:

- Recipient: first line of text,
- Sender: second line of text,
- Message: third and remaining lines of text
- All: the entire contents of the file

Literal search text special characters:

- ~ as first character means the start of the variable's contents
- ~ as last character means the end of variable's contents
- ? matches a single character
- * matches anything (including nothing)
- \ following 3 digits are a decimal character value
- the ~, ?, *, and \ characters must use \ with decimal equivalent to be used

literally

RegEx replacement:

Backreferences are allowed in replacement text (i.e. \1 \2 \3 ...)

\r, \n, and \### (3 digits) literals are supported for inserting CR, LF, and

ASCII

Literal search text variables

%Recipient% - the current value for Recipient

%Sender% - the current value for Sender

%Message% - the current value for Message

%All% - the current value for All of the text

%Temp1-10% - the current value for that Temp variable

\ following 3 digits are a decimal character value

Literal replacement text variables:

%Recipient% - the current value for Recipient

%Sender% - the current value for Sender

%Message% - the current value for Message

%All% - the current value for All of the text

%Match% - the actual text that matched the search text (Literal or RegEx match types)

%Group1-10% - RegEx group backreferences obtained from a RegEx search

%Temp1-10% - the current value for that Temp variable

%InputFileName% - the name of the input file being processed

%Year%, %Month%, %Day%, %MHour%, %Hour%, %Minute%, %Second%, %MSecond%, %AMPM%

%Unique% - unique number - guaranteed unique to the computer - pretty unique otherwise

\ following 3 digits are a decimal character value

Functions:

Format is %FunctionName(input-value)% - it will be replaced with the function's output value

%DateTimeToUnixTime()%

The default value of the Temp1-10 variables are an empty string

The default value of the Flag variable is false

Comments: The <Comment> section can be used in each <Filter> and it will be output

in the debug log, or any line of text with a pound sign (#) as the very first character (no leading spaces or tabs) will be considered a comment, and ignored,

and also not logged

How PageGate Accepts Messages

2

The Recipient, Sender, Message and ALL variables are dynamically linked together. So, for

example, if you change the Sender, the first line of All will also be updated.

Changing

All will update the Recipient, Sender and Message parts

When the order of things matter:

<Filter> block are executed in the order they appear (unless <Goto> is used)

In a <Filter> block:

<Search> (if it exists) runs first and determines if the <MatchFound> or <MatchNotFound> block is run

<Replace> <WriteFile> <Run> <RunFilter> <ReadFile> <GoTo>

<Abort> <Continue> run in that order

There can be multiple <Replace> <WriteFile> <Run> <RunFilter> <ReadFile> blocks and they will run

in the order they appear - i.e. all <Replace> blocks will run, then all <WriteFile> blocks, etc.

RegEx uses a Pearl compatible implementation syntax

good reference: <http://www.regular-expressions.info/tutorialcnt.html>

Breaking changes

<Section></Section> is renamed to <Variable></Variable>

In the <Replace> section the <All></All> tag has been renamed to

<ReplaceAll></ReplaceAll>

In the <Replace> <Scope> section the value: 'Section' is renamed to 'Entire'

All exiting <Replace> section have to be surrounded with <MatchFound></MatchFound> tags

Recipient, Sender, Message and All are new dynamically linked together

Bare Filter Template - for easier cutting/pasting

* in sections with multiple value options, the first in the list is the default value

<InputFile>

<Name>

Parameter

</Name>

<Delete>

False, True

</Delete>

</InputFile>

<OutputFile>

```
<Name>
    InputFile
</Name>
</OutputFile>
<Debug>
    <Console>
        <Active>
            False, True
        </Active>
    </Console>
    <File>
        <Active>
            False, True
        </Active>
        <Name>

        </Name>
        <Append>
            False, True
        </Append>
    </File>
    <LogInputFile>
        False, True
    </LogInputFile>
    <LogOutputFile>
        False, True
    </LogOutputFile>
</Debug>
<Filter>
    <Comment>

    </Comment>
    <FilterName>

    </FilterName>
    <Lookup>
        <Variable>
            All, Recipient, Sender, Message, Temp1-10
        </Variable>
        <CaseSensitive>
            False, True
        </CaseSensitive>
        <SpaceSensitive>
            False, True
```

```
</SpaceSensitive>
<List>

  </List>
</Lookup>
<Search>
  <Type>
    Literal, RegEx, GreaterThan, LessThan
  </Type>
  <Variable>
    All, Recipient, Sender, Message, Temp1-10, Flag
  </Variable>
  <Text>

    </Text>
</Search>
<MatchFound>
  <Replace>
    <Type>
      Literal, RegEx, And, Or
    </Type>
    <Variable>
      All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Scope>
      Match, Entire
    </Scope>
    <ReplaceAll>
      False, True
    </ReplaceAll>
    <Text>

      </Text>
    </Replace>
  <WriteFile>
    <Name>

      </Name>
    <Variable>
      All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
  </WriteFile>
  <Run>
    <Command>
```

```
</Command>
<Async>
    True, False
</Async>
<Show>
    False, True
</Show>
</Run>
<RunFilter>
    <SettingsFile>

    </SettingsFile>
    <Variable>
        , All, Recipient, Sender, Message, Temp1-10
    </Variable>
    <InputFile>

    </InputFile>
    <DebugFile>

    </DebugFile>
    <AppendDebug>
        False, True
    </AppendDebug>
    <UseDll>
        True, False
    </UseDll>
</RunFilter>
<ReadFile>
    <Name>

    </Name>
    <Variable>
        All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Delete>
        False, True
    </Delete>
</ReadFile>
<Goto>

</Goto>
<Abort>
    False, True
```

```
</Abort>
<Continue>
    True, False
</Continue>
<MatchFound>
<MatchNotFound>
<Replace>
    <Type>
        Literal, And, Or
    </Type>
    <Variable>
        All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Text>

    </Text>
</Replace>
<WriteFile>
    <Name>

    </Name>
    <Variable>
        All
    </Variable>
</WriteFile>
<Run>
    <Command>
    </Command>
    <Async>
        True, False
    </Async>
    <Show>
        False, True
    </Show>
</Run>
<RunFilter>
    <SettingsFile>

    </SettingsFile>
    <Variable>
        , All, Recipient, Sender, Message, Temp1-10
    </Variable>
    <InputFile>
```

```
</InputFile>
<DebugFile>

</DebugFile>
<AppendDebug>
    False, True
</AppendDebug>
<UseDll>
    True, False
</UseDll>
</RunFilter>
<ReadFile>
    <Name>

    </Name>
    <Variable>
        All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Delete>
        False, True
    </Delete>
</ReadFile>
<Goto>

</Goto>
<Abort>
    False, True
</Abort>
<Continue>
    True, False
</Continue>
<MatchNotFound>
</Filter>
```

Filter

The Filter section determines what you want to do with the data read as input. A single filter script may reference multiple individual <Filter> sections to perform multiple tasks within the same data set. You can also find [filter statement examples here](#) and [complete script examples here](#).

```
<Filter>
    <Search>
        See Search section
    </Search>
```


How PageGate Accepts Messages

2

<Replace>
 See [Replace](#) section.
</Replace>
<Continue>
 See Values and Description
</Continue>
<Abort>
 See Values and Description
</Abort>
</Filter>

Flag	Values and Description
Search	Please review the Search section for a detailed explanation of this tag and its components.
Replace	Please review the Replace section for a detailed explanation of this tag and its components.
Continue	This tag determines whether or not you want the script to continue to the next action if it finds a match. This tag can be set to a value of True or False. For example, let's say we're searching for the text 345 within the input file's data. Let's also say that no match for 345 can be found in the input data. If Continue is set to True, the filter will move to the next specified <Filter> section. If Continue is set to False, the script will immediately terminate upon not finding a match.
Abort	If the search section finds a match and Abort has a value of True, the script is aborted. Basically, the input data is left unchanged and no more filter blocks are processed. It just abruptly ends the filter processing and quits out without saving anything.

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<u>IMPORTANT</u>

	<p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Search

The Search section determines what you want the filter to look for.

```

<Search>
  <Section>
    See Values and Description
  </Section>
  <Type>
    See Values and Description
  </Type>
  <Text>
    See Values and Description
  </Text>
</Search>
    
```

Flag	Values and Description	
Section	This tag determines where the filter should look for the data specified in the Text tag (see below).	
	Value s	Explanation
	All	All tells the filter to search the entire data set read from the input file.
	Recipi ent	Recipient tells the filter to search only in the first line of the file.
	Sende r	Sender tells the filter to search only in the second line of the file.
	Messa ge	Message tells the filter to search only in the third and following lines of the file.
	Temp	Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags. See Example 1 and Example 3 .
Flag	Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for	

	advanced logic chains. See Example 1 and Example 3 .
Type	<p>Values: Literal or RegEx</p> <p>This tag determines whether you want to use literal text or a RegEx statement in the Text tag.</p>
Text (Literal)	<p>If the Type tag is set to Literal, this tells the filter to search on the text entered in the Text tag. For example, you could enter the literal text Fire Department if you wanted the filter to key off of the literal text "Fire Department" (without the quotes).</p> <p>This tag supports the following boolean parameters:</p> <ul style="list-style-type: none"> * - Multiple Character Wildcard For example, fire* would find fire, firewood, fireplace, fireman and similar. ? - Single Character Wildcard For example, fire? would find fires, fired but would not find fire. ~ as first character means start of section/file ~ as last character means end of section/file For example, ~12345~ would only read 12345. Without the ~ before and after the string, the filter would recognize 12345 in 0123456. <p>This tag also supports the following variables:</p> <ul style="list-style-type: none"> %Recipient% - the current value for Recipient %Sender% - the current value for Sender %Message% - the current value for Message %All% - the current value for All of the text %Temp% - the current value for the Temp variable \ - following 3 digits are a decimal character value. See Below.
Text (Regex)	<p>The PageGate Filter script supports RegEx expressions. RegEx uses a Perl compatible implementation syntax and a good reference guide for the language can be found here: http://www.regular-expressions.info/tutorialcnt.html</p>

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.

?	Single character wildcard.
*	Multiple character wildcard.
IMPORTANT	
\	<p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>

Replace

The Replace section determines what you want to do with the data found in the [Search](#) section.

<Replace>

<Section>

See Values and Description

</Section>

<Type>

See Values and Description

</Type>

<Scope>

See Values and Description

</Scope>

<All>

See Values and Description

</All>

<Text>

replacement text goes here

must be quoted to retain beginning and/or ending whitespace

See Values and Description

</Text>

</Replace>

Flag	Values and Description	
	Values: All, Recipient, Sender, Message, Temp, Flag	
	This tag determines which section the filter perform the replacement within.	
Section	Value	Explanation
	All	All tells the filter to replace any instance of the searched text within the entire data set read from the input file.

	Recipient	Recipient tells the filter to replace values only in the first line of the file.
	Sender	Sender tells the filter to replace values only in the second line of the file.
	Message	Message tells the filter to replace values only in the third and following lines of the file.
	Temp	Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags.
	Flag	Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains.
Type		<p>Values: Literal, RegEx, And, Or</p> <p>This tag determines whether you want to use literal text or a RegEx statement in the Text tag. This tag can also be used as a situational boolean operator to determine how to proceed.</p> <p>The boolean values (Or & And) modify the Flag value. So, for example, you could have the filter search for a value and set it as the Flag, then have the replace filter use "Or" to indicate that the Flag value may be more than one thing. You could also use the And operator to tell the filter that the Flag value must contain multiple variables to be matched. See Example 1 and Example 3.</p>
Scope		Values: Match or Section
All		Values: True or False
Text (Literal)		<p>If the Type tag is set to Literal, this tells the filter to search on the text entered in to the Text tag. For example, you could enter the literal text Fire Department if you wanted the filter to key off of the literal text "Fire Department" (without the quotes).</p> <p>This tag supports the following boolean parameters:</p> <ul style="list-style-type: none"> * - Multiple Character Wildcard For example, fire* would find fire, firewood, fireplace, fireman and similar. ? - Single Character Wildcard For example, fire? would find fires, fired but would not find fire. <p>~ as first character means start of section/file ~ as last character means end of section/file</p>

How PageGate Accepts Messages

2

	For example, ~12345~ would only read 12345. Without the ~ before and after the string, the filter would recognize 12345 in 0123456.
Text (Reg ex)	The PageGate Filter script supports RegEx expressions. RegEx uses a Pearl compatible implementation syntax and a good reference guide for the language can be found here: http://www.regular-expressions.info/tutorialcnt.html

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output. This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.

Filter Statement Examples

To better demonstrate how some of the flags function, here are a few examples of complete filter statements and an explanation of what they do.

Here is an example of a [Literal Text Replacement Command](#).

Here is an example of an [Abort Command](#).

Abort Command

This Filter statement tells the script to terminate if the [Flag](#) variable is set to False:

```
<Filter>
  <Search>
    <Section>
      Flag
    </Section>
```

```
<Type>
  Literal
</Type>
<Text>
  False
</Text>
</Search>
<Replace>
  <Section>
    Flag
  </Section>
  <Scope>
    Section
  </Scope>
  <Type>
    And
  </Type>
  <Text>
    False
  </Text>
  <All>
    False
  </All>
</Replace>
<Continue>
  False
</Continue>
</Filter>
```

Literal Text Replacement Command

This Filter statement tells the script to replace one piece of literal text with another. More specifically, this statement tells the script to search for the literal text Boiler Room and replace it with boiler_tech_group.

```
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Boiler Room
```

```
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            boiler_tech_group
        </Text>
        <All>
            False
        </All>
    </Replace>
</Filter>
InputFile
```

The InputFile section determines which file(s) you want the filter to scan.

```
<InputFile>
    <Name>
        See Values and Description
    </Name>
    <Delete>
        See Values and Description
    </Delete>
</InputFile>
```

Flag	Values and Description
Name	Values: Parameter or Static Input File Name This tag determines the name(s) of the file(s) that you want the filter to modify. Just to note, the value Parameter is literal and not figurative. To use this value, you'll want to use the text Parameter, including the capital P. When the filter is engaged by the pstproc.bat file, it is fed the file name being used by the filter. The Parameter value tells the filter that it should simply accept the file name as it was passed by the batch file.

	Alternatively, you can specify a static file for input. For example, c:\PageGateData\Filter\winsck9000.log would tell the filter to only read from a specific file named winsck9000.log. As another example, you could specify c:\PageGateData\Filter*.log and that would tell the filter to read all log files placed in the directory.
Delete	<p>Values: True or False</p> <p>This tag determines whether you want the filter to delete the input file when it has finished processing it. If you do, set this tag to True. If you do not, set this tag to False.</p>

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>

OutputFile

The OutputFile section determines what you want to name the files output by the filter's modifications. This section is defined as follows:

```
<OutputFile>
  <Name>
    See Values and Description
  </Name>
</OutputFile>
```

Flag	Values and Description
Name	<p>Values: InputFile or Static Input File Name</p> <p>This tag determines the names of the files that you want the filter to output after it has modified the data read from the input file.</p> <p>The value InputFile (this is case sensitive) tells the filter that the file output should mirror the original input file's name. So, for example, if the file read was named example.txt, then the file output would also be called example.txt</p> <p>Alternatively, you can specify a static naming convention for the output files. This value also supports the following variables:</p> <p>%InputFileName% %InputFileNameBase% %InputFileNameExtension% %InputFilePathAndName% %InputFilePath%</p> <p>So, for example, you could specify this as the value: c:\PageGateData\ASCII%InputFileNameBase%.asc</p> <p>That would tell the filter that it should use the original name of the file read but the output file should have the extension .asc instead of the extension on the original file.</p>

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>

How PageGate Accepts Messages

2

Debug

The Debug section determines whether you want a debug level review of the filter to appear before the filter completes its actions. This is incredibly useful when first configuring a script and we recommend that this flag be enabled until the filter is ready to put in to production. This section is defined as follows:

```
<Debug>  
    See Values and Description  
</Debug>
```

Flag	Values and Description
Debug	Values: True or False This tag determines whether you want to review the debug level information for the filter before it completes its actions. Set this flag to True to review the debugging information, set this flag to False if you do not.

Please see [Debugging the Script](#) for more information.

Complete Script Examples

Due to the complex nature of the Filter script, we've created a few examples of how scripts can be implemented. This is by no means a complete list of everything that the script can do and is intended to be a set of examples to guide you through the possibilities the script offers.

[Example 1](#)

In this example, we've configured the filter script to search the message read from the input file for the text BIOHAZARD and replace it with bio_hazard. We're then inserting the static text WARNING! EXTREME HAZARD DETECTED: before the matched search string.

[Example 2](#)

In this example, we've configured the filter to scan through XML tags for the text Technical or Network in the <AlertType> tag, then pull in information from the <Incident> and <Description> tags. This collated data is then being passed to a statically named group in PageGate: techsupervisors.

[Example 3](#)

In this example, we've configured the filter to scan for certain keywords and key phrases. If it finds them, messages containing that text should not be allowed to go through and should instead be filtered.

[Example 4](#)

In this example, we've configured the filter script for a dispatching agency that needs all fire, rescue and accident calls dispatched but does not want non-emergency calls dispatched. To do this, the script reads in the entire input file and looks for certain flags like FIRE,DISPATCHED or RESCUE,DISPATCHED or ACCID,DISPATCHED.

If it finds any of those three strings, the script then begins massaging and reformatting the data to remove unwanted information and to determine who the message should go to and what the message should say.

In this particular example, all fire and rescue calls need to be delineated from each other so that the fire calls go to the fire department and the rescue calls go to the rescue department but all accident calls should go to both fire and rescue departments.

[Example 5](#)

In this example, we're using an [HTTPGET](#) carrier to poll data from the National Weather service to have it delivered as an SMS.

[Example 6](#)

This example is designed to work with an [SMS Replies](#) system and is configured to look for the words 'subscribe' and 'unsubscribe' in the body of reply messages. If it finds 'subscribe', it reconfigures the message to go to the recipient or group in PageGate named 'subscribe'. If it finds 'unsubscribe', it reconfigures the message to go to the recipient or group named 'unsubscribe'.

[Example 7](#)

This example is designed to work with an [SMS Replies](#) system and is configured so that any time someone replies to a message with 'unsubscribe' or 'subscribe', those replies execute a batch file or script contained within a batch file. This can be especially useful if you're using an SQL environment and have PageGate's database linked to it with ODBC expressions. In theory, you could code a script to be executed that would automatically remove an entry from the users table of the database any time the 'unsubscribe' script was executed.

[Example 8](#)

This example is designed to work with an [SMS Replies](#) system. This script is configured with a list of 'allowed' phone numbers and is also configured to scan the sender value of all replies. If the sender's phone number isn't in the 'allowed' list, the message is filtered as spam.

Example 1

In this example, we've configured the filter script to search the message read from the input file for the text 'BIOHAZARD'. If it is found then the script will replace the current Recipient with 'bio_hazard' to indicate that all messages containing the word BIOHAZARD should go to the bio_hazard group. We're then inserting the static text 'WARNING! EXTREME HAZARD DETECTED:' before the matched search string.

How PageGate Accepts Messages

2

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

This section of the filter configures the script to look for a static file named Example.txt in the same directory as the script and, when the script completes, it will not delete the original file and will output to a static file named Output.txt to the directory in which the script resides.

```
<InputFile>
  <Name>
    Example.txt
  </Name>
  <Delete>
    False
  </Delete>
</InputFile>

<OutputFile>
  <Name>
    Output.txt
  </Name>
</OutputFile>

<Debug>
  False
</Debug>

# This section of the filter configures the script to search
# for BIOHAZARD and if found, replaces the Recipient
# with bio_hazard.
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      BIOHAZARD
    </Text>
  </Search>
  <Replace>
```

```
<Section>
    Recipient
</Section>
<Type>
    Literal
</Type>
<Scope>
    Section
</Scope>
<Text>
    bio_hazard
</Text>
</Replace>
<Continue>
    True
</Continue>
</Filter>
```

This section of the filter configures the script to insert
WARNING! EXTREME HAZARD DETECTED: before the BIOHAZARD text

```
<Filter>
    <Search>
        <Section>
            Message
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            BIOHAZARD
        </Text>
    </Search>
    <Replace>
        <Section>
            Message
        </Section>
        <Type>
            Literal
        </Type>
        <Scope>
            Match
        </Scope>
        <Text>
            WARNING! EXTREME HAZARD DETECTED: %Match%
```

How PageGate Accepts Messages

2

```
</Text>
</Replace>
</Filter>
Original Content
```

This is an example of the input file for [filter example 1](#). This is what it looks like before the filter engages.

```
supervisors
AlarmMonitor
BIOHAZARD ROOM 354
```

Another example:

```
it_alerts
AlarmMonitor
BIOHAZARD RULES IN EFFECT FOR WING 2
Filtered Content
```

When [the data from example 1](#) is put through [the example 1 filter](#), this is the output:

```
bio_hazard
AlarmMonitor
WARNING! EXTREME HAZARD DETECTED: BIOHAZARD ROOM 354
```

or like this:

```
bio_hazard
AlarmMonitor
WARNING! EXTREME HAZARD DETECTED: BIOHAZARD RULES IN EFFECT FOR
WING 2
```

Example 2

In this example, we've configured the filter to scan through an XML file for the text 'Technical' or 'Network' in the <AlertType> tag. It will then temporarily store the information from the <Incident> and <Description> tags in the Temp variable. This collected data is then being passed to a statically named group in PageGate: techsupervisors.

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

```
<InputFile>
  <Name>
    Parameter
  </Name>
  <Delete>
    True
  </Delete>
</InputFile>
```

How PageGate Accepts Messages

```

<Debug>
    False
</Debug>

<OutputFile>
    <Name>
        c:\PageGateData\ASCII\%InputFileNameBase%.asc
    </Name>
</OutputFile>

```

This section of the Filters tells the script to look for
any lines that have <AlertType>Technical</AlertType> and
set the Flag variable to True if it is found

```

<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            <AlertType>Technical</AlertType>
        </Text>
    </Search>
    <Replace>
        <Section>
            Flag
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Or
        </Type>
        <Text>
            True
        </Text>
    </Replace>
</Filter>

```

This section of the Filters tells the script to look for any
lines that have the <AlertType>Network</AlertType> and
set the Flag variable to True if it is found


```
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      <AlertType>Network</AlertType>
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Or
    </Type>
    <Text>
      True
    </Text>
  </Replace>
</Filter>
```

This Filter section tells the script to abort if no instances of
the two search strings can be found:

```
<Filter>
  <Search>
    <Section>
      Flag
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      False
    </Text>
  </Search>
  <Replace>
    <Section>
```

```
        Flag
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        And
    </Type>
    <Text>
        False
    </Text>
</Replace>
<Continue>
    False
</Continue>
</Filter>
```

This section of the filter tells the script to grab data
contained in the <Incident> tag.

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            <Incident>*</Incident>
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            %Match%
```

```
        </Text>
    </Replace>
</Filter>
```

This section of the filter tells the script to grab data from
the <Description> tag

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            <Description>*</Description>
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            %Temp% %Match%
        </Text>
    </Replace>
</Filter>
```

This section of the filter removes the tag
<Incident> from the message.

```
<Filter>
    <Search>
        <Section>
            Temp
        </Section>
        <Type>
            Literal
```

```
        </Type>
        <Text>
            <Incident>
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Match
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>

        </Text>
    </Replace>
</Filter>
```

This section of the filter removes the tag
</Incident> from the message.

```
<Filter>
    <Search>
        <Section>
            Temp
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            </Incident>
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Match
        </Scope>
        <Type>
```

```
        Literal
    </Type>
    <Text>

    </Text>
</Replace>
</Filter>
```

This section of the filter removes the tag
<Description> from the message.

```
<Filter>
  <Search>
    <Section>
      Temp
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      <Description>
    </Text>
  </Search>
  <Replace>
    <Section>
      Temp
    </Section>
    <Scope>
      Match
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>

    </Text>
  </Replace>
</Filter>
```

This section of the filter removes the tag
</Description> from the message.

```
<Filter>
  <Search>
```

```
<Section>
    Temp
</Section>
<Type>
    Literal
</Type>
<Text>
    <Incident>
</Text>
</Search>
<Replace>
    <Section>
        Temp
    </Section>
    <Scope>
        Match
    </Scope>
    <Type>
        Literal
    </Type>
    <Text>
</Text>
</Replace>
</Filter>
```

This Filter tells the script to output the file in the standard
PageGate GetASCII file format for direct processing:

```
<Filter>
    <Search>
        <Section>
            Flag
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            True
        </Text>
    </Search>
    <Replace>
        <Section>
            All
```

How PageGate Accepts Messages

2

```
</Section>
<Scope>
    Section
</Scope>
<Type>
    Literal
</Type>
<Text>
    TechSupervisors\013\010CAD\013\010%Temp%
</Text>
</Replace>
</Filter>
```

Original Content

This is example XML content to be processed by [filter example 2](#). This is what it looks like before the filter engages.

```
<XML>
<AlertType>Technical</AlertType>
<Requester>Station 3</Requester>
<Assets>PC01, PC02</Assets>
<Category>Desktop Support</Category>
<Incident>Replace Computers</Incident>
<Description>Workstations 01 and 02 should be replaced on Thursday</Description>
</XML>
```

Another example:

```
<XML>
<AlertType>Network</AlertType>
<Requester>Admin Group</Requester>
<Assets>Router</Assets>
<Category>Network Support</Category>
<Incident>DNS Failure</Incident>
<Description>DNS failures detected in node 2</Description>
</XML>
```

Filtered Content

When [the data from filter example 2](#) is processed by [filter example 2](#), this is the output content:

```
techsupervisors
CAD
Replace Computers, Workstations 01 and 02 should be replaced on Thursday
```

Another example:

techsupervisors

CAD

DNS Failure, DNS failures detected in node 2

Example 3

In this example, we're scanning for keywords and keyphrases that indicate a message shouldn't be sent. If the filter finds any of the keywords or keyphrases, it will filter those messages from going through. For this example, we've configured the filter to look for the key phrases "Return from Alarm" and "Alarm Acknowledged" as well as potential partial matches.

```
<InputFile>
  <Name>
    Parameter
  </Name>
  <Delete>
    True
  </Delete>
</InputFile>

<Debug>
  True
</Debug>

<OutputFile>
  <Name>
    c:\PageGateData\output\%InputFileNameBase%.asc
  </Name>
</OutputFile>

<Filter>
<Comment>look for Return From Alarm</Comment>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Return From Alarm
    </Text>
  </Search>
<Replace>
```


How PageGate Accepts Messages

2

```
<Text>
</Text>
</Replace>
<Abort>
True
</Abort>
</Filter>

<Filter>
<Comment>look for Partial Phrases of Return From Alarm</Comment>
<Search>
  <Section>
    All
  </Section>
  <Type>
    Literal
  </Type>
  <Text>
    Retu*
  </Text>
</Search>
<Replace>
<Text>
</Text>
</Replace>
<Abort>
True
</Abort>
</Filter>

<Filter>
<Comment>look for Alarm Block Acknowledged</Comment>
<Search>
  <Section>
    All
  </Section>
  <Type>
    Literal
  </Type>
  <Text>
    Alarm Block Acknowledged
  </Text>
</Search>
<Replace>
```

```
<Text>
</Text>
</Replace>
<Abort>
  True
</Abort>
</Filter>

<Filter>
<Comment>look for Partial "Alarm Block Acknowledged" phrases</Comment>
<Search>
  <Section>
    All
  </Section>
  <Type>
    Literal
  </Type>
  <Text>
    Block
  </Text>
</Search>
<Replace>
<Text>
</Text>
</Replace>
<Abort>
  True
</Abort>
</Filter>
```

Example 4

Example 3

In this example, we've configured the filter script for a dispatching agency that needs all fire, rescue and accident calls dispatched but does not want non-emergency calls dispatched. To do this, the script reads in the entire input file and looks for certain keywords like FIRE,DISPATCHED or RESCUE,DISPATCHED or ACCID,DISPATCHED.

If it finds any of those three strings, the script then reformats the data, removes unwanted information, and determines who the message should be sent to.

In this particular example, all fire and rescue calls need to be separated from each other so that the fire calls go to the fire department and the rescue calls go to the rescue department, but all accident calls should go to both fire and rescue departments.

How PageGate Accepts Messages

2

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

This section of the filter configures the script to accept whatever file name is passed by the preproc.bat file, to delete the input file when it's finished and to output the reformatted data to the c:\PageGateData\ASCII\ folder with a file name mirroring the original file but with a .asc extension instead of the original file's extension.

```
<InputFile>
  <Name>
    Parameter
  </Name>
  <Delete>
    True
  </Delete>
</InputFile>

<Debug>
  False
</Debug>

<OutputFile>
  <Name>
    c:\PageGateData\ASCII\%InputFileNameBase%.asc
  </Name>
</OutputFile>
```

The next three sections of the Filters tells the script to look for any lines that have
the text ,FIRE,DISPATCHED| or ,RESCUE,DISPATCHED| or ACCID,DISPATCHED| in
the input file

and sets the temporary Flag variable to True if any are found

look for lines with station FIRE,DISPATCHED identifier

```
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,FIRE,DISPATCHED|
```

```
        </Text>
    </Search>
    <Replace>
        <Section>
            Flag
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Or
        </Type>
        <Text>
            True
        </Text>
    </Replace>
</Filter>

# look for lines with station RESCUE,DISPATCHED identifier
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            ,RESCUE,DISPATCHED|
        </Text>
    </Search>
    <Replace>
        <Section>
            Flag
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Or
        </Type>
        <Text>
            True
        </Text>
```

How PageGate Accepts Messages

2

```
</Replace>
</Filter>

# look for lines with station ACCID,DISPATCHED identifier
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,ACCID,DISPATCHED|
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Or
    </Type>
    <Text>
      True
    </Text>
  </Replace>
</Filter>

# This section test the value of the Flag variable
# and aborts out of the script if it is not set to True
# Note that the values in the <Replace> section really
# don't matter because the script is aborting

# abort if FIRE/RESCUE/ACCID DISPATCHED not found
<Filter>
  <Search>
    <Section>
      Flag
    </Section>
    <Type>
```

```
        Literal
    </Type>
    <Text>
        False
    </Text>
</Search>
<Replace>
    <Section>
        Flag
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        And
    </Type>
    <Text>
        False
    </Text>
</Replace>
<Continue>
    False
</Continue>
</Filter>
```

Now that the filter has determined that there is relevant data
in the message file, these filter statements will search for
department names and store what is found in the Temp variable

look for lines with station F-BAY identifier

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            /F-BAY
        </Text>
    </Search>
    <Replace>
        <Section>
```

```
        Temp
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        Literal
    </Type>
    <Text>
        F-BAY
    </Text>
</Replace>
</Filter>

# look for lines with station F-STU identifier
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            /F-STU
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            F-STU
        </Text>
    </Replace>
</Filter>
```

This Filter section tells the script to abort if no

How PageGate Accepts Messages

acceptable station is found within the data

(the Temp variable is still blank)

```
<Filter>
  <Search>
    <Section>
      Temp
    </Section>
  </Search>
  <Type>
    Literal
  </Type>
  <Text>
    </Text>
  </Text>
</Search>
<Replace>
  <Section>
    Flag
  </Section>
  <Scope>
    Section
  </Scope>
  <Type>
    And
  </Type>
  <Text>
    False
  </Text>
</Replace>
<Continue>
  False
</Continue>
</Filter>
```

This Filter section tells the script to grab a portion of the data read for modification:

It grabs everything between the first character of the line and ,DISPATCHED| keyword

and removes everything else

```
<Filter>
  <Search>
    <Section>
      All
    </Section>
  </Search>
  <Type>
    Literal
  </Type>
```



```
<Text>
    *,DISPATCHED|
</Text>
</Search>
<Replace>
    <Section>
        All
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        Literal
    </Type>
    <Text>
        %Match%
    </Text>
</Replace>
</Filter>
```

This Filter flag takes the data read by the previous statement and modifies it:
It gets rid of extra stuff between date/time and address and replaces it with
a carriage return and line feed

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            (Eastern Daylight Time)*!%Temp%,F,
        </Text>
    </Search>
    <Replace>
        <Section>
            All
        </Section>
        <Scope>
            Match
        </Scope>
        <Type>
            Literal
```

```
        </Type>
        <Text>
            \013\010
        </Text>
    </Replace>
</Filter>
```

The next two filter sections separate the rescue calls from the fire calls
so that they go to two different agencies/organizations/units.

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            FIRE
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            %Temp%-FIRE
        </Text>
    </Replace>
</Filter>
```

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
```

```
        </Type>
        <Text>
            RESCUE
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            %Temp%-RESCUE
        </Text>
    </Replace>
</Filter>
```

The next three Filter sections remove the text ,DISPATCHED| from the message
and replaces it with more readable text

get rid of DISPATCHED on RESCUE

```
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            ,RESCUE,DISPATCHED|
        </Text>
    </Search>
    <Replace>
        <Section>
            All
        </Section>
        <Scope>
            Match
        </Scope>
```

How PageGate Accepts Messages

```

    <Type>
      Literal
    </Type>
    <Text>
      \013\010RESCUE
    </Text>
  </Replace>
</Filter>

# get rid of DISPATCHED on FIRE
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,FIRE,DISPATCHED|
    </Text>
  </Search>
  <Replace>
    <Section>
      All
    </Section>
    <Scope>
      Match
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      \013\010FIRE
    </Text>
  </Replace>
</Filter>

# get rid of DISPATCHED on ACCID
<Filter>
  <Search>
    <Section>
      All
    </Section>

```

How PageGate Accepts Messages

```

        <Type>
            Literal
        </Type>
        <Text>
            ,ACCID,DISPATCHED|
        </Text>
    </Search>
    <Replace>
        <Section>
            All
        </Section>
        <Scope>
            Match
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            \013\010ACCID
        </Text>
    </Replace>
</Filter>

```

This Filter section tells the script to output the file in the standard
 # PageGate GetASCII file format for direct processing:

convert to standard .asc format

```

<Filter>
    <Search>
        <Section>
            Flag
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            True
        </Text>
    </Search>
    <Replace>
        <Section>
            All
        </Section>
        <Scope>

```

```
        Section
    </Scope>
    <Type>
        Literal
    </Type>
    <Text>
        %Temp%\013\010CAD\013\010Dispatch %Temp%\013\010%All%
    </Text>
</Replace>
</Filter>
```

Original Content

This is example content to be processed by [filter example 3](#). This is what it looks like before the filter engages.

```
2018-08-23 14:07:30.307 (Eastern Daylight Time) CISCOUNTSTAT||T=1|
V=6233M070737305319||T=4|V=02||T=0|V=U,00/F-BAY,P,PRIMARY STREET/CROSS
STREET,INJ ACCID,DISPATCHED||
```

Filtered Content

When [the data from filter example 4](#) is processed by [filter example 4](#), this is the output content:

```
F-BAY
DISPATCH
P,PRIMARY STREET/CROSS STREET,INJ ACCID
```

Example 5

This filter script is designed to work with the National Weather service's XML content, which is generated by polling their web API.

```
<Debug>
    <Console>
        <Active>True</Active>
    </Console>
    <File>
        <Active>True</Active>
        <Name>weatherfilter.txt</Name>
        <Append>False</Append>
    </File>
    <LogInputFile>True</LogInputFile>
    <LogOutputFile>True</LogOutputFile>
</Debug>

<Filter>
<Comment>Delete if no severe weather alerts</Comment>
    <Search>
```

How PageGate Accepts Messages

2

```
<Type>Literal</Type>
<Variable>All</Variable>
<Text><title>There are no active watches, warnings or
advisories</title></Text>
</Search>
<MatchFound>
  <Replace>
    <Type>Literal</Type>
    <Variable>All</Variable>
    <Scope>Entire</Scope>
    <Text></Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Check to see if this is the raw weather data or the ascii formatted data. If the
latter, skip to Process.</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>Sender</Variable>
    <Text>WeatherSend</Text>
  </Search>
  <MatchFound>
    <Goto>Process</Goto>
  </MatchFound>
</Filter>

<Filter>
<Comment>Turn each entry block in to a separate file.</Comment>
<FilterName>EntryLoop</FilterName>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(?s)(<entry>.*?</entry>)</Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>Temp8</Variable>
      <Scope>Entire</Scope>
      <Text>YourRecipientOrGroup\013\010WeatherSend\013\010%
Group1%</Text>
    </Replace>
    <WriteFile>
```

```
<Variable>Temp8</Variable>
<Name>INSERT THE PATH TO YOUR WEATHER FOLDER\posts\
%InputFileNameBase%-%FilterIndex%.asc</Name>
  </WriteFile>
</MatchFound>
<MatchNotFound>
  <Replace>
    <Variable>All</Variable>
    <Scope>Entire</Scope>
    <Text></Text>
  </Replace>
  <Goto>Finished</Goto>
</MatchNotFound>
</Filter>

<Filter>
<Comment>Get rid of the entry block processed, then loop back until there are no more
entry blocks</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(s)<entry>.*?</entry></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
    <Goto>EntryLoop</Goto>
  </MatchFound>
  <MatchNotFound>
    <Goto>Finished</Goto>
  </MatchNotFound>
</Filter>

<Filter>
<FilterName>Process</FilterName>
<Comment>Grab cap:effective and put it in Temp4</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><cap:effective>(.*?)</cap:effective></Text>
  </Search>
```



```
<MatchFound>
  <Replace>
    <Variable>Temp4</Variable>
    <Scope>Entire</Scope>
    <Text>%Group1%</Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Grab cap:expires and put it in Temp5, then put the final ID in
Temp6</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><cap:expires>(.*?)</cap:expires></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>Temp5</Variable>
      <Scope>Entire</Scope>
      <Text>%Group1%</Text>
    </Replace>
    <Replace>
      <Variable>Temp6</Variable>
      <Scope>Entire</Scope>
      <Text>%Temp4%%Temp5%</Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
  <Comment> Turn Temp6 in to a valid file name and put it in Temp7. It will be used
to read/write the requestId files</Comment>
  <Search>
    <Type>RegEx</Type>
    <Variable>Temp6</Variable>
    <Text>[^a-zA-Z0-9\-.]</Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>RegEx</Type>
      <Variable>Temp7</Variable>
      <Scope>Entire</Scope>
```

```
<Text>%Group1%</Text>
<ReplaceAll>True</ReplaceAll>
</Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Pull in the ID from the file</Comment>
<MatchNotFound>
<ReadFile>
<Variable>Temp9</Variable>
<Name>INSERT THE PATH TO YOUR WEATHER FOLDER\ids\%
Temp7%.asc</Name>
</ReadFile>
</MatchNotFound>
</Filter>

<Filter>
<Comment>Check to see if the ID already exists. If it does, delete everything. If it
doesn't, write the ID file</Comment>
<Search>
<Type>Literal</Type>
<Variable>Temp9</Variable>
<Text>%Temp7%</Text>
</Search>
<MatchFound>
<Replace>
<Variable>All</Variable>
<Scope>Entire</Scope>
<Text></Text>
</Replace>
<Goto>Finished</Goto>
</MatchFound>
<MatchNotFound>
<WriteFile>
<Variable>Temp7</Variable>
<Name>INSERT THE PATH TO YOUR WEATHER FOLDER\ids\%
Temp7%.asc</Name>
</WriteFile>
</MatchNotFound>
</Filter>

<Filter>
<Comment>Remove from cap:event to cap:certainty</Comment>
```

How PageGate Accepts Messages

```

<Search>
  <Type>Regex</Type>
  <Variable>All</Variable>
  <Text>(s)<cap:event>.*?</cap:certainty></Text>
</Search>
<MatchFound>
  <Replace>
    <Type>Regex</Type>
    <ReplaceAll>True</ReplaceAll>
    <Variable>All</Variable>
    <Scope>Match</Scope>
    <Text></Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Remove from entry to author</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(s)<entry>.*?</author></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <ReplaceAll>True</ReplaceAll>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Replace cap:areaDesc with LOCATION</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>All</Variable>
    <Text><cap:areaDesc></Text>
  </Search>
  <MatchFound>
    <Replace>
      <ReplaceAll>True</ReplaceAll>

```

```
<Variable>All</Variable>
<Scope>Match</Scope>
<Text>\013\010LOCATION:\032</Text>
</Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Remove the XML tags from the link</Comment>
<Search>
  <Type>Regex</Type>
  <Variable>All</Variable>
  <Text><link href="(.)"V></Text>
</Search>
<MatchFound>
  <Replace>
    <Variable>All</Variable>
    <Scope>Match</Scope>
    <Text>%Group1%</Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Remove from cap:polygon to cap:entry</Comment>
<Search>
  <Type>Regex</Type>
  <Variable>All</Variable>
  <Text>(s)<cap:polygon>.*?<Ventry></Text>
</Search>
<MatchFound>
  <Replace>
    <Type>Regex</Type>
    <ReplaceAll>True</ReplaceAll>
    <Variable>All</Variable>
    <Scope>Match</Scope>
    <Text></Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Remove the summary</Comment>
<Search>
```

```
<Type>Regex</Type>
<Variable>All</Variable>
<Text><summary>.*</summary></Text>
</Search>
<MatchFound>
  <Replace>
    <Variable>All</Variable>
    <Scope>Match</Scope>
    <Text></Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Remove the opening title tag</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><title></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <Variable>All</Variable>
      <ReplaceAll>True</ReplaceAll>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>
<Filter>
<Comment>Remove the closing title tag</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text></title></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <Variable>All</Variable>
      <ReplaceAll>True</ReplaceAll>
      <Scope>Match</Scope>
      <Text></Text>
```

```
        </Replace>
      </MatchFound>
    </Filter>

  <Filter>
  <Comment>Remove the closing cap:areaDesc tag</Comment>
    <Search>
      <Type>Regex</Type>
      <Variable>All</Variable>
      <Text></cap:areaDesc></Text>
    </Search>
    <MatchFound>
      <Replace>
        <Type>Regex</Type>
        <Variable>All</Variable>
        <ReplaceAll>True</ReplaceAll>
        <Scope>Match</Scope>
        <Text></Text>
      </Replace>
    </MatchFound>
  </Filter>

  <Filter>
  <Comment>Clean up the text, part 1</Comment>
    <Search>
      <Type>Literal</Type>
      <Variable>All</Variable>
      <Text>\013\010\013\010</Text>
    </Search>
    <MatchFound>
      <Replace>
        <ReplaceAll>True</ReplaceAll>
        <Variable>All</Variable>
        <Scope>Match</Scope>
        <Text>\013\010</Text>
      </Replace>
    </MatchFound>
  </Filter>

  <Filter>
  <Comment>Processing completed</Comment>
  <FilterName>Finished</FilterName>
    <MatchNotFound>
    </MatchNotFound>
```

How PageGate Accepts Messages

2

</Filter>

Example 6

Example 5

This filter script is designed to work with an [SMS Replies](#) system and is configured to look for the words 'subscribe' and 'unsubscribe' in the body of reply messages. If it finds 'subscribe', it reconfigures the message to go to the recipient or group in PageGate named 'subscribe'. If it finds 'unsubscribe', it reconfigures the message to go to the recipient or group named 'unsubscribe'.

Note: The <InputFile> and <OutputFile> sections are not included, so the script will use the filename passed in by PageGate, and write the script's results back to the same filename when done.

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

look for 'unsubscribe' in message text (all lower case)

if found, route the message to 'unsubscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      unsubscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      unsubscribe-recipient-or-group-here
    </Text>
  </Replace>
  <Continue>
    False
```

```
</Continue>
</Filter>

# look for 'Unsubscribe' in message text (begins with capital)
# if found, route the message to 'unsubscribe' recipient
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Unsubscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      unsubscribe-recipient-or-group-here
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

# look for 'subscribe' in message text (all lower case)
# if found, route the message to 'subscribe' recipient
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
```



```
        subscribe
      </Text>
    </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      subscribe-recipient-or-group-here
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>
```

look for 'Subscribe' in message text (begins with capital)

if found, route the message to 'subscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      subscribe-recipient-or-group-here
    </Text>
  </Replace>
```

```
<Continue>
    False
</Continue>
</Filter>
```

Example 7

This filter script is designed to work with an [SMS Replies](#) system and is configured so that any time someone replies to a message with 'unsubscribe' or 'subscribe', those replies will be sent to an 'unsubscribe' or 'subscribe' recipient in PageGate that will execute a batch file or script contained within a batch file.

To do so, there's a bit of preparation you'll need to do to tell PageGate what script to run. It's also important to note that this batch file can be used to execute powershell commands.

Step 1: Create a directory for your batch file

- 1) Create a directory for the batch file(s) to reside in. This will also be the output folder for the carrier we'll create in Step 2. For example, c:\PageGateData\unsubscribe\

Step 2: Create a FILE Carrier

- 1) Open PG Admin.
- 2) Right click Carriers, select Add.
- 3) Give the carrier a name, something straightforward like BATCH.
- 4) Set Protocol to: FILE
- 5) Set File Path to the directory you created in Step 1 above. By default, c:
 \PageGateData\unsubscribe\

Step 3: Create a recipient named 'unsubscribe' and tie it to the FILE carrier

- 1) Right click Recipients, select Add.
- 2) Name the recipient: unsubscribe
- 3) Set Carrier to the FILE carrier you created in step 2. By default, BATCH.
- 4) Click Apply.

Step 4: Place and edit the appropriate script files

- 1) Open the directory you created in step 1 above. By default, c:
 \PageGateData\unsubscribe\
- 2) Copy pstproc.bat from the PageGate program directory in to that folder.
- 3) Edit pstproc.bat to execute the desired effect.

How PageGate Accepts Messages

2

Step 5: Configure the SMS Replies Filter System

look for 'unsubscribe' in message text

if found, route the message to 'unsubscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      unsubscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      unsubscribe-recipient
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>
```

look for 'Unsubscribe' in message text

if found, route the message to 'unsubscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Unsubscribe
    </Text>
```

```
</Search>
<Replace>
  <Section>
    Recipient
  </Section>
  <Scope>
    Section
  </Scope>
  <Text>
    unsubscribe-recipient
  </Text>
</Replace>
<Continue>
  False
</Continue>
</Filter>

# look for 'subscribe' in message text
# if found, route the message to 'subscribe' recipient
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      subscribe-recipient
    </Text>
  </Replace>
  <Continue>
    False
```

```
</Continue>
</Filter>

# look for 'Subscribe' in message text
# if found, route the message to 'subscribe' recipient
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      subscribe-recipient
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>
```

Example 8

This example is designed to work with an [SMS Replies](#) system. This script is configured with a list of 'allowed' phone numbers and is also configured to scan the sender value of all replies. If the sender's phone number isn't in the 'allowed' list, the message is filtered as spam.

```
<Filter>
  <Comment>
    This filter is scanning the sender value for a series of phone numbers. In this example,
    we're using a simple Regex string to create an 'or' statement. So, we're scanning for the
    first number or the second number or the third number and so on.
  </Comment>
```

```
<Search>
  <Section>
    Sender
  </Section>
  <Type>
    Regex
  </Type>
  <Text>
    (phonenumber1|phonenumber2|phonenumber3|phonenumber4|etc)
  </Text>
</Search>
```

```
<Replace>
<Comment>
```

If the filter finds any of the specified phone numbers, it assigns the intended delivery group or recipient.

```
</Comment>
  <Section>
    Recipient
  </Section>
  <Scope>
    Section
  </Scope>
  <Text>
    name of messaging group or recipient
  </Text>
</Replace>
<Continue>
  False
</Continue>
</Filter>
```

GetMail Integration

[The GetMail API](#) allows you to receive SMTP/Email traffic and translate it into outbound messages in a different format. For example, you could email a message to PageGate to deliver as a message to cell phones, other email addresses, pagers, web APIs, etc. Enabling GetMail allows you to have all traffic to a recipient or group forwarded to the specified email address as well as allowing PageGate to host a POP accessible mailbox on the PageGate server for that recipient or group.

PageGate has a fully functional mail server engine as a core part of the program but without the GetMail module, you only ever access the outbound functions of this mail server. The GetMail module unlocks the receiving portion of our mail server engine.

GetMail can receive email/SMTP messages in one of two ways:

- Sub-domain hosting
- POP Mailbox

How PageGate Accepts Messages

2

The best method of email integration is to have PageGate host a sub-domain off of your existing domain. To do this, you'll want to talk to your mail server administrator and ask them to create this sub-domain for you, then redirect the MX record for that sub-domain to the internal IP address of the PageGate server. For example, let's say that you have an email domain of whatever.com (and that your email addresses are something@whatever.com). You could have messaging.whatever.com created and redirected to PageGate's IP. As soon as you do, every recipient and group in the program becomes a valid email address.

So, let's say that we have a recipient called sales and we have GetMail configured to use messaging.whatever.com. You could send an email to sales@messaging.whatever.com and that email would come in to PageGate, then get processed as an outbound message for the recipient named sales.

POP is a relatively antiquated method of receiving email, many modern mail servers don't support it. However, if your mail server does still support POP mailboxes, you can configure GetMail to periodically check those email addresses, then read in any email waiting in the mailbox and deliver that traffic to a recipient or group of your specification.

With the ubiquity of email, more and more modern applications are integrating the ability to send email. If your application has the ability to send email, it can easily be integrated with PageGate by using [PageGate's email interface](#). It is worth noting that this interface can be used to integrate applications running on different platforms (Unix, Mac, etc.). For more information, please refer to the [GetMail](#) section of this manual for more details.

GetWeb Integration

[The GetWeb API](#) provides a CGI executable that can be hosted on a web server and a series of pre-formatted templates to publish a messaging website or series of messaging websites. Web Developers can use this web server interface to send messages to PageGate from their webpages (HTML, ASP, JavaScript, etc.). This means that you can host messaging websites on your local intranet or, if GetWeb is interfaced with an outward facing web server, on the internet.

GetWeb's CGI executable can be used with either a POST or GET method.

The POST method can be used by creating a form on your webpage and pointing to the webgate.exe program in the FORM ACTION statement.

The GET method is more popular with scripting languages such as JavaScript and Java. It allows the GetWeb CGI to be called with a command similar to one that could be entered into a browser's URL field.

The GetWeb interface can be used to enable PageGate to be integrated with applications running on different platforms (Unix, Mac, etc.). Please refer to the [GetWeb](#) section of this manual for more details.

You can also find examples of Linux and Powershell scripts that can be used with GetWeb [in our documentation here](#).

GetSerial Integration

[The GetSerial API](#) allows PageGate to monitor a serial/RS232 port, virtual or physical, for raw data text strings to be parsed in to outbound messages.

When a data string is sent to monitored a serial port, GetSerial pattern matches sections of the data string to parse information for messaging. For more information, please see the [GetSerial](#) portion of the documentation.

GetTAP Integration

[The GetTAP API](#) allows PageGate to monitor a serial port or answer a modem call to receive TAP transmitted pages.

This allows PageGate to function as a TAP receiver or a TAP terminal for any devices and systems that can transmit that protocol, such as a Zetron terminal. For more information, please see the [GetTAP](#) portion of the documentation.

If you need to monitor more than just one serial port or phone line for incoming TAP messages, you can do so with PageGate's TAP2Ascii module. For more information, please contact our Sales Department at 781-829-0500 xt1 or at sales@notepage.com

SNPP Server Integration

The SNPP API allows PageGate to function as an SNPP server to receive this protocol from a variety of sources. Messages received by this API are output to a directory as a text file that is then read in with the GetASCII API. As a result, this API requires the GetASCII API to function.

To configure the PageGate SNPP server:

- 1) Create a directory where the SNPP Server and its files will reside. We recommend creating the following directory: C:\PageGateData\SNPP\
- 2) In Windows, browse in to the PageGate installation directory.
- 3) Move the snppserver.exe, pininfo.txt and snpp.ini files to the folder you created in step 1.
- 4) The first thing we'll need to do is modify the snpp.ini file. Open the file in notepad.

To give you a little more information, this .ini file controls the behavior of the PageGate SNPP Server. If you are running the evaluation version of the PageGate SNPP Server, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

- 3) Change the OutputFolder= value to the directory you created in step 1.

How PageGate Accepts Messages

2

4) Change the LogFolder= value to your PageGate Logging directory. If you are unsure of what this folder is, open the PageGate Admin and go to the Program - Settings section.

5) Save the snpp.ini file and close notepad.

=====

1) Open pininfo.txt in notepad.

2) This file tells the SNPP server what range of numbers can be passed to the server. The format of the entries defined in this text file are:

"Starting PIN", "Ending PIN", "recipient type"

If you would like ID numbers in a range to be recognized as the recipient's name (not using ad-hoc, but using numeric recipient names in PageGate), use the text "PG_Recipient" as shown below:

"0001", "9999", "PG_Recipient"

If you would like ID numbers in a range to be passed through PageGate as intact ID's to another source in an ad-hoc range, use the name of a Recipient in PageGate whose type is Ad-Hoc.

"0000001", "4999999", "adhoc1"

=====

To set the SNPP Server to run as a Windows Service, do the following:

1) Open a command prompt.

2) Change directory to the SNPP Server's program directory.

3) Type the following and hit enter:
snppserver /install

4) Exit the command prompt.

5) Open the Services list in Windows.

6) Go in to the Properties of the PageGate SNPP Server service.

7) Change the Startup Type to Automatic and Start the Service.

Once this has been done, the final step is to configure the GetASCII module to read the data from the SNPP server. To do so:

1) Open the PageGate Admin.

2) Go to Interfaces - GetASCII - Settings.

- 3) In the Polling Directory, specify the PageGate SNPP Server's output folder.
- 4) Put a check in Enabled.
- 5) Click Apply.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the SNPP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc

TouchTone Receiver Integration

This module provides the ability to monitor a voice modem to answer calls and receive DTMF signals and output them as text to a file.

To configure the PageGate Touch Tone Receiver:

- 1) Create a directory where the SNPP Server and its files will reside. For example, the directory:
C:\TouchTone\
 - 2) In Windows, browse in to the PageGate installation directory.
 - 3) Move the pggttone.ini and pgttreceiver.exe files to the folder you created in step 1.
 - 4) The first thing we'll need to do is modify the pggttone.ini file. Open the file in notepad.

To give you a little more information, this .ini file controls the behavior of the PageGate Touch Tone Receiver. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

- 3) Change RunStyle=Application to RunStyle=Service
- 4) Change SwitchStyle=False to SwitchStyle=True
- 5) Set the DataFolder= value to your GetASCII polling directory. If you do not have GetASCII enabled, please see the [GetASCII](#) section of the documentation.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the SNPP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc

- 6) Set the ComPort= value to match the COM port of your modem.
Note: Only modems that have the 'voice' capability can receive DTMF signals. Modems that do not have that ability cannot interpret them and cannot be used for this purpose.

How PageGate Accepts Messages

2

7) For the Initialization and Connection strings, please use the settings relevant to your modem as detailed in the file.

8) Save the file and close notepad.

9) Open a command prompt.

10) Change directory to the directory you created in step 1.

11) Type the following and hit enter:

```
pgtreceiver /install
```

12) Exit the command prompt.

13) Open the Services list in Windows.

14) Go in to the Properties of the PageGate TouchTone service.

15) Change the Startup Type to Automatic and Start the Service.

Direct Database Access

Developers that are familiar with database access can directly integrate with PageGate's database.

Note: Please consider other integration methods first. Writing improperly formatted lines to the database will interfere with PageGate's stability. Empty fields **MUST** be expressed as blank string values and **CANNOT** be left null. Each field value must exactly match the expected format.

This is probably the most efficient way to add messages to the message queue, but may not be an option in all programming environments. Another downside to this method is that it may be version specific (if the database structure changes in a future version of PageGate, the integration may stop working).

PageGate uses a Jet 4.0 database as it's back end. This is a MS Access 2000 compatible database. If you have the ability to write to MS Access 2000 databases, then you can write to the PageGate's database directly. New messages can be added to the PageGate's message queue by adding a new record to the 'messages' table in the 'pagegate.mdb' database.

[Click here for field descriptions and values in the Messages table](#)

[Click here for field descriptions and values in the Users and Groups table](#)

How PageGate Accepts Messages

2

The Messages Table

The field definitions, descriptions and values for the 'Messages' table.

Warning: Empty fields should be represented as blank string values. NULL values will not be properly be parsed.

Label	Field Type	Value
user	Text	This field must match the name of a Recipient or Group within the PageGate Admin.
pin	Text	Must be set to a zero length string (""). This field value CANNOT be left Null.
message	Memo	This field is the text of the message to be sent
carrier	Text	Must be set to a zero length string (""). This field value CANNOT be left Null.
date_time_requested	Date	This field specifies the date and time to send this message. By default, set the current date and time.
date_time_sent	Date	Leave unset
frm	Text	This field specifies the name of the Sender of the message.
sent	Boolean	Set to False
error	Boolean	Set to False
ascii_file	Text	Must be set to a zero length string (""). This field value CANNOT be left Null.
mail_file	Text	Must be set to a zero length string (""). This field value CANNOT be left Null.
lock	Boolean	Set to False
scheduled	Boolean	Set to False
reliapage_index	Numeric	Set to 0
repeat_interval	Numeric	Set to 0
repeat_stop	Date	This value should be set to the same value as the date_time_requested field.
station	Numeric	This field provides a unique identifier that can be referenced when performing reports to identify the source of the message. Values 0 through 9 are system reserved and cannot be used. Set this field value to 10 ≥

message_id	Auto Number	Leave unset
retries	Numeri c	Set to 0
lasty_try	Date	This value should be set to the 30 seconds before the date_time_requested field.
transactio n	Numeri c	<u>REQUIRED VALUE</u> This field must be a unique number (long integer). The best way to handle this is to edit the control database record, read the current value of the 'unique' field, increment it, and then write this value back to the control table. The transaction number can be reused after approximately 1 hour.
dialer	Numeri c	Set to 0
creator_name	Text	If security is enabled within PageGate, this is the name of the user who created the message. Otherwise, set the field value to a zero length string (""). This field value CANNOT be left Null.
creator_machine	Text	This is the machine name of the computer that created the message. This field value CANNOT be left Null.
original_recipient	Text	Set this field value to the same value as the user field.
last_recipient	Text	Set this field value to the same value as the user field.
server	Text	This field is not used but must be set to a zero length string (""). This field value CANNOT be left Null.
archived	Boolea n	Set to False.
priority	Numeri c	Set this field value to 0.

The Users and Groups Table

Field definitions, descriptions and values for the 'Users' table in the database.

Warning: Empty fields should be represented as blank string values. NULL values will not be properly be parsed.

Label	Field Type	Description and Potential Values
user	Text	The internal database name of a recipient or group and is the value referenced by the GetASCII, GetWeb, GetTAP and GetSerial interfaces. For instance, this field name is the value referenced by network monitoring utilities, CAD dispatching, ticket tracking systems and similar.

full_name	Text	The recipient or group's display name in the PageGate GUI Client and in any messaging websites this recipient or group appears on.
id	Text	The ID to message. This will vary widely based on what type of messages this recipient will receive. It could be a 10 or 11 digit ID for SMS, it could be an email address, it could be a custom ID string or it could be something else entirely.
EmailTo	Text	When this recipient is messaged through SMTP based carriers, this is the email address that will be messaged. This field overrides what may be in the ID field but only for delivery via the SMTP protocol, otherwise the ID field is respected. It is possible to tie a recipient to a non-SMTP method of delivery, like cellular hardware or an SMS delivery service (AWSSNS, Twilio, etc), then configure a failover scenario where that non-SMTP method of delivery fails over to SMTP. When that happens, this field's delivery value will be used instead of the id value.
carrier	Text	This value must exactly match the name of a carrier as it exists in the 'carrier' field of the Carriers table in the database; which is also the name displayed in the 'Carriers' section of PG Admin.
maxchars	Numerical	The maximum number of characters this recipient should have the ability to receive. This value may be higher than the carrier's 'Max Chars' value. If the recipient value is higher than the carrier value, PageGate will break up long messages in to multiple data blocks based on the carrier value.
on_call	Boolean	0 for False. -1 for True. This field should be set to False for recipients. If a group should have an on-call schedule, set this field to True. Schedules are configured in PG Admin or by modifying values in the table 'oncalltable' of the database.
reliapage_index	Numerical	Set this field to 0.
forward	Text	If you own the GetMail interface and would like a copy of all messages sent to this recipient or group forwarded to an email address, specify that email address in this field. If you do not own the GetMail interface, leave this field as a blank string value.
ascii_service	Boolean	0 for False. -1 for True. If GetAscii is part of your license, set to True. Otherwise, set to False.
gui_service	Boolean	0 for False. -1 for True.

How PageGate Accepts Messages

2

		If the PageGate GUI Client is part of your license, set to True. Otherwise, set to False.
mail_service	Boolean	0 for False. -1 for True. If GetMail is part of your license, set to True. Otherwise, set to False.
serial_service	Boolean	0 for False. -1 for True. If GetSerial is part of your license, set to True. Otherwise, set to False.
tap_service	Boolean	0 for False. -1 for True. If GetTAP is part of your license, set to True. Otherwise, set to False.
web_service	Boolean	0 for False. -1 for True. If GetWeb is part of your license, set to True. Otherwise, set to False.
ad_hoc	Boolean	0 for False. -1 for True. This field should be set to False for groups. If this recipient is an ad-hoc recipient, set to True. Otherwise, set to False.
multi_page	Boolean	0 for False. -1 for True. This field should be set to False for recipients. If this group is tied to the multi-page GetWeb template, set this to True. Otherwise, set to False.
email_only	Boolean	0 for False. -1 for True. This field should be set to False for groups. If this recipient should only exist as a mailbox for use with GetMail , set to True. Otherwise, set to False.
group	Boolean	0 for False. -1 for True. This field should be set to False for recipients. If this entry should be a group , set to True. Otherwise, set to False.
display_group	Boolean	0 for False. -1 for True. This field should be set to False for recipients and groups. If this entry should be a display group , set to True. Otherwise, set to False.
template	Text	Set to a blank string value.

text1 - text10	Text	Text fields that are referenced by GetWeb's basic templates and values that may be called on by PageGate's template system. In most cases, leave these as blank string values.
image1 - image5	Text	Text fields that are referenced by GetWeb's basic templates and values that may be called on by PageGate's template system. In most cases, leave these as blank string values.
custom	Boolean	0 for False. -1 for True. This field should be set to False for recipients. If this group is tied to a custom web template, set to True. Otherwise, set to False.
group_list	Boolean	0 for False. -1 for True. This field should be set to False for recipients. If this group is tied to a web template and the members of this group should display on the website it generates, set to True. Otherwise, set to False.
GLFullName	Boolean	0 for False. -1 for True. This field should be set to False for recipients. If this group_list is true, setting this field to True will configure the website to reference the value in the full_name field instead of the recipient field. In all other cases, set to False.
Failover	Text	This field may only be set for recipients and has no effect on groups. The value of this field must exactly match the name of a recipient or group as it appears in the user field. If messages to this recipient fail, they will be redirected to the specified value. Otherwise, leave this field as a blank string value.
extra1 - extra4	Text	Text fields that are referenced by the PageGate GUI Client and values that may be called on by PageGate's template system. In most cases, leave these as blank string values.
hold	Boolean	0 for False. -1 for True. This field should be set to False for groups.

		<p>If this recipient should have a hold placed on their messages during specified dates and times, set to True and configure the values in hold_start and hold_stop.</p> <p>In all other cases, set to False.</p>
hold_start	Date/Time	<p>If hold is set to True, enter the current date and what time the message hold should begin. If hold is set to False, enter the current date and time. The field value won't be referenced if hold is False but this field can't be left blank.</p> <p>Date values should be expressed as m/dd/yyyy and Time values should be expressed as HH:MM:SS AM/PM.</p> <p>Example: 7/16/2020 8:35:00 PM</p> <p>The only part of the field that will remain relevant is the time value specified. PageGate will automatically update the date value to the current date if hold is True.</p>
hold_stop	Date/Time	<p>If hold is set to True, enter the current date and what time the message hold should end. If hold is set to False, enter the current date and time. The field value won't be referenced if hold is False but this field can't be left blank.</p> <p>Date values should be expressed as m/dd/yyyy and Time values should be expressed as HH:MM:SS AM/PM.</p> <p>Example: 7/16/2020 7:35:00 AM</p> <p>The only part of the field that will remain relevant is the time value specified. PageGate will automatically update the date value to the current date if hold is True.</p>
creator	Text	<p>This field specifies the name of an administrator or administrative group that should have creator/control rights to this recipient or group. If you aren't using the security system, leave this field as a blank string value.</p> <p>Otherwise, match this value to the name of an administrator or administrative group in your PageGate security list.</p>
priority	Numeric	<p>Set to 100 for groups.</p> <p>Can be set to 1 - 32000 for recipients and reflects that recipient's priority in messaging. Recipients with values closer to 1 are sorted higher in the delivery order while recipients with values closer to</p>

		32000 are sorted lower. For more information, see Prioritizing Messages .
--	--	-------------------------------------------------------------------------------------------

PageGate has the ability to send messages in a variety of formats to a variety of sources.

Among other methods, you can:

- [Send SMS/text messages to phones](#)
- [Send pages to pagers](#)
- [Send Email](#)
- [Make API calls to Web API, REST API, or web form](#)
- [Use XMPP chat clients](#)
- [Send faxes](#)
- [Send RAW data via TCP or Serial](#)
- [Trigger scripts, including Powershell](#)

Since reliability is incredibly important, PageGate has a [failover](#) system that allows you to automatically fail over from one method of communication to another in the event of failure.

For more information on how to configure a fail over, have a look at the [Failover Scenarios](#) section of the documentation.

SMS (text message) Delivery Methods

Sending SMS is relatively easy and it can be done in a few ways, some of which are hardware based and some of which are internet based.

Typically speaking, hardware methods of SMS delivery tend to be more reliable as they do not rely on an internet connection but SMS delivery services and carrier enterprise systems can be incredibly reliable methods of delivery.

- [Carrier Public Gateways](#)

Most carriers support an email-to-SMS system as a free, public service and PageGate supports the ability to send SMTP/Email messages . We try to keep an up to date listing of [known carrier domains on our support site](#).

However, due to the nature of email delivery, this method can be somewhat unreliable and delayed. While this is a viable method of SMS delivery, we don't recommend using it as the primary method.

For more information, see the [Carrier Public Gateways](#) section of the documentation.

- [Cellular Hardware](#)

How to Send Messages

3

On the hardware side of things, there are cellular modems and gateways. These devices have an interface PageGate can use to send and receive SMS just like a cell phone does.

Cellular modems are easier to use with physical systems as they connect by USB and/or RS-232.

Cellular gateways are easier to use in virtual environments as they can be reached via TCP by Telnet or SSH.

For more information, including make and model recommendations, have a look at the [Using Cellular Hardware](#) section of the documentation.

- **[General SMS delivery services](#)**

SMS delivery services are companies that provide a messaging API for you to deliver traffic through and you typically license your SMS services through them, though not every SMS delivery service is intended for use in every environment.

For a list of known SMS delivery services, have a look at the [Message Aggregators section of our integration page](#).

General messaging services are designed to be accessible to anyone and are incredibly easy to register and configure for the end user, though they are typically intended for lower volume environments. These services include Amazon Web Services SNS, Twilio, MessageBird and similar.

Some companies provide SMS services as an integrated part of a telephony or VOIP solution and PageGate can easily trigger messages through their API. These services include API Daze, Teli and similar.

High volume messaging services are designed for implementations that will need to send 100,000+ messages per month. Contracts are negotiated with the high volume service and PageGate uses their API to trigger messages. These services include Bandwidth, Teli, Infobip and similar.

For more information, have a look at the [General SMS Delivery Services](#) section of the documentation.

- **[Public Safety delivery services](#)**

Some SMS delivery services are designed to work with public safety dispatching to provide mapping, geospatial and other key data to first responders. These services include Active911, I Am Responding, CodeMessaging and similar.

For more information, have a look at the [Public Safety Delivery Services](#) section of the documentation.

- **[Medical Industry delivery services](#)**

Some SMS delivery services are designed to work with hospitals, clinics and other medical services to provide HIPAA compliant secure messaging. These services include TigerConnect, QliqSOFT, Voalte and similar.

For more information, have a look at the [Medical Industry Delivery Services](#) section of the documentation.

- **[Carrier Enterprise Systems](#)**

The major North American carriers each have their own respective enterprise messaging platforms. Verizon has EMAG (Enterprise Messaging Access Gateway), AT&T has FirstNet, Sprint has EMG (Enterprise Messaging Gateway), and so on.

Note for Public Safety: Verizon offers free access to dispatch related public safety organizations if you ONLY message Verizon phones through EMAG. Enabling inter-carrier operability requires a paid contract.

PageGate supports the protocols necessary to use these platforms to send your SMS.

For more information, see the [Carrier Enterprise Systems](#) section of the documentation.
Carrier Public Gateways

Most carriers support an email-to-SMS system as a free, public service and PageGate supports the ability to send SMTP/Email messages. We try to keep an up to date listing of [known carrier domains on our support site](#).

However, due to the nature of email delivery, this method can be somewhat unreliable and delayed. While this is a viable method of SMS delivery, we don't recommend using it as the primary method due to the potential delays and missed messages inherent to it.

PageGate has the ability to send SMTP/Email messages in one of two ways:

- Direct

Direct delivery configures PageGate to use its own mail server engine to directly negotiate with the outside world.

To configure this method of delivery, you will need to open port 25 and add the public IP address of your PageGate server to the SPF record for your email domain.

- Relayed

Relayed delivery configures PageGate to pass all of its traffic through an SMTP server.

To configure this method of delivery, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Technically, direct delivery is more efficient and reliable because it removes at least one hop in the delivery sequence and also means that PageGate isn't reliant on another mail server for delivery. However, when you use direct delivery for SMTP/Email, you'll need abide by all of the rules required of a mail server because the program will be functioning as a mail server to deliver SMTP/Email on behalf of your email domain.

Due to some of the complications involved in configuring PageGate for direct delivery, it may be more practical to configure PageGate to relay its SMTP/Email traffic through your SMTP/Email server. However, doing so makes PageGate reliant on that mail server for message delivery.

Step 1: Configure the Carrier as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:
Carrier: The name of the carrier.
Protocol: SMTP
Host: The domain (part after the @) of the carrier host. For a list of known hosts, please see the [SMTP Delivery](#) section of our site.
Port: When relaying SMTP, specify the port required by your SMTP server. When using direct SMTP, set the port to 25.
Max Chars: 160
- 4) Go to Carriers - YourNewCarrier - Advanced
- 5) Specify the following:
When relaying SMTP, select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.
For example, if you use Office365, you would use smtp.office365.com as the relay server address.
Specify the required username and password.

When using direct SMTP, select 'Deliver directly to destination server'.
- 6) Click Apply
- 7) Go to Carriers - YourNewCarrier - Template
- 8) Specify the following in the 'Settings' section:
From = The value from the username field
EXAMPLE:
From = [CAD@notepage.com](#)

9) Click Apply.

Step 2: Create Recipients tied to the Carrier

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: The recipient's cell phone carrier

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Cellular Hardware

Cellular hardware is the one of the most reliable and efficient method of delivering SMS. By using cellular hardware, you're removing PageGate's reliance on your internet connection to deliver your messages which means that your internet connection could go down and your messages would still be delivered. It's also important to note that allowing PageGate to interact with cellular hardware is the a method of configuring a [two-way messaging system](#) with SMS.

SMS capable cellular hardware comes in two varieties: cellular gateway/router and cellular modem.

Cellular Gateways/Routers

Cellular routers and gateways can be accessed via TCP by Telnet or SSH. You don't have to connect these devices to a PC, they just need to be connected to your network and configured appropriately to receive messaging commands.

Good examples of cellular routers/gateways are:

- Multitech's MTR-LNA7
- Airlink's GX450, RV50, RV55 and LX40
- Microhard IP4nGii
- Cisco Series 800, 900

How to Send Messages

3

- Cradlepoint AER1600/1650 Series, AER31x0, AER2100, AER2200 Series, COR IBR200, COR IBR600B/C Series, COR IBR900 Series, COR IBR1100 Series, COR IBR1700, MBR1400, MBR1200B, ARC CBA850, ARC CBA750B*, IBR300, IBR6x0, IBR11x0

For information on how to configure TCP accessible cellular hardware, have a look at the [Creating a GSM-AT-IP Carrier](#) steps.

Cellular Modems

A cellular modem connects to a PC by USB or RS232 and provides a COM port in the operating system.

If PageGate is running in a virtual environment, you can also connect a cellular modem to a COM port virtualizer that provides the COM port to the virtual machine. This grants PageGate access to the cellular modem, which allows it to send and receive text messages just like a cell phone does.

Good examples of cellular modem are:

- Multitech's MTC-LNA4
- Microhard's IP4NG
- AirPrime 73xx and 75xx
- Skyus-DS
- BlueOcean GSM and GPRS series
- Duxbury 120 and 180
- Siemens MTC45MC, MTC45TC, TC35, TC65
- Wavecom Fastrack GSM Modems

For information on how to configure RS232 or USB connected cellular hardware, have a look at the [Creating a GSM-AT Carrier](#) steps.

General SMS Delivery Services

For a list of known SMS delivery services, have a look at the [Message Aggregators section of our integration page](#).

SMS delivery services are companies that provide a messaging API for you to deliver traffic through and you typically license your SMS services through them, though not every SMS delivery service is intended for use in every environment.

Most SMS delivery services offer at least one API that PageGate can use to send messages and some services offer multiple methods that can be used for failover and

redundancy. PageGate supports the following delivery methods for internet or local network SMS delivery services:

- HTTP POST and HTTP GET

These methods are used to interact with Web and REST APIs, whether those are internet or local network based. Any SMS service or application that can accept a web form, POST or GET methods can be used for messaging with this protocol.

Typically speaking, there will be a few variables needed to set things up; things like account number, API key and authentication credentials

Examples:

- [AWS SNS](#)
- [Twilio](#)
- [MessageBird](#)
- [API Daze](#)
- [Telnyx](#)
- [Voximplant](#)

- SMTP

Some delivery services offer the ability to email messages in to their platform can can be used for messaging with this protocol.

Services that support this delivery method will have an email domain you'll deliver your traffic through. For configuration steps, please see the [Creating a Relayed SMTP Carrier](#) section.

If your SMS delivery service supports multiple API hosts or delivery methods, you can configure them for failover and redundancy. For more information, see the [Failover Scenarios](#) section of the documentation.

For a list of known SMS delivery services, have a look at the [Message Aggregators section of our integration page](#).

Configuring AWS SNS

Step 1: Create an Amazon Web Services Account

- 1) Go to <https://aws.amazon.com/>
- 2) Click an AWS Create Account
- 3) Enter the email address and password.

- 4) Enter payment information (required by Amazon)
- 5) Choose your support plan.

Step 2: Create an Access Key for PageGate to use

Creating an access key can be done by providing PageGate access to the root account and password or you can tie the AWS SNS permissions to a specific user account..

If you want PageGate to use the root account, you would do the following:

- 1) Log in to your AWS account.
- 2) In the upper right hand corner, click on your account's name and select 'My Security Credentials'
- 3) Click 'Create new Access Key'
- 4) Copy the 'Access Key ID' and 'Secret Access Key' values as we'll need both in PageGate.

If you want to grant the ability to use AWS SNS to an existing account, you would do the following:

- 1) Log in to your AWS account.
- 2) In the upper right hand corner, click on your account's name and select 'My Security Credentials'
- 3) On the left hand side, click 'Users'
- 4) Select the user in question.
- 5) Click 'Add Permissions'
- 6) Select 'Attach existing policies directly'
- 7) In the 'Filter policies' field, enter: SNS
- 8) Check 'AmazonSNSFullAccess'
- 9) Click Next.
- 10) Click 'Add Permissions'
- 11) Select the 'Security Credentials' tab for that user.
- 12) Click 'Create Access Key'
- 13) Copy the 'Access Key ID' and 'Secret Access Key' values as we'll need both in PageGate.

Step 3: Configure AWS SNS in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:

Carrier: AWSSNS

Protocol: AwsSns

Host: [Please see Amazon's SNS documentation](#)

<https://docs.aws.amazon.com/sns/latest/dg/sns-supported-regions-countries.html>

The 'Endpoint' value is what you will use in the this field.

Examples:

sns.us-east-1.amazonaws.com

sns.us-west-2.amazonaws.com

Max Chars: 160

- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - AWSSNS - Advanced
- 6) Enter the Access Key ID value in the Username field.
- 7) Enter the Secret Access Key value in the Password field.
- 8) Click Apply.

Step 4: Create Recipients tied to the AWSSNS method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: AWSSNS

Failover: See [Failover Scenarios](#)

ID/PIN: cell phone number, including country code.

NOTE: For messaging in North America, it is imperative that you provide the 1 in front of the phone number. AWS SNS treats the first character of the phone number as the

How to Send Messages

3

country code and you can unintentionally send international SMS if you do not provide the 1.

<u>Correct</u>	<u>Incorrect</u>
17818290500	781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring Twilio

Step 1: Create a Twilio Account

- 1) Go to <https://www.twilio.com/>
- 2) Click Sign Up
- 3) Create your Twilio Account

Step 2: Create an Access Key for PageGate to use

- 1) Log in to your Twilio account.
- 2) On the left hand side, click Settings, then API Keys.
- 3) Click 'Create new API Key'
- 4) In 'Friendly Name', enter: PageGate
- 5) Leave 'Key Type' set to: Standard
- 6) Click 'Create API Key'
- 7) Copy the SID and SECRET values as we'll need both for PageGate.

Step 3: Configure Twilio in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:

Carrier: Twilio

Protocol: HTTPPOST

Host: api.twilio.com/2010-04-01/Accounts/YourAccountSID/Messages

NOTE: Your account SID is part of the host address.

Port: 443

Max Chars: 160

- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - Twilio - Advanced
- 6) Enter the SID in 'Username'.
- 7) Enter the SECRET value in 'Password'.
- 8) Click Apply.
- 9) Go to Carriers - Twilio - Template
- 10) Specify the following in the 'Text' section and click Apply:

|To|`%RecipientId%`|

|From|The Phone Number of your Twilio Account in international format. Example:
+17818290500|

|Body|`%Message%`|

Step 4: Create Recipients tied to the Twilio method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: Twilio

Failover: See [Failover Scenarios](#)

ID/PIN: cell phone number, including country code.

Correct	Incorrect
17818290500	781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

- 3) Click Apply.

Configuring MessageBird

Step 1: Create a MessageBird Account

- 1) Go to <https://messagebird.com/en/>
- 2) Click Sign Up

How to Send Messages

- 3) Create your MessageBird Account
- 4) In 'Select an interface' section, select REST APIs
- 5) In 'Select a product to try first' section, select SMS API
- 6) In 'Select a use case', select Send SMS
- 7) In 'Get your free test credits', enter your cell phone number for confirmation.
- 8) Enter your verification code.
- 9) Click 'Skip'

Step 2: Create an Access Key for PageGate to use

- 1) Log in to your MessageBird account.
- 2) On the right side, under Your API Keys, click Manage API Keys.
- 3) On the left side, click API access
- 4) Click 'Show Key' next to the account you want PageGate to use.
- 5) Copy the Key.

Step 3: Configure MessageBird in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:
 - Carrier:** MessageBird
 - Protocol:** HTTPPOST
 - Host:** rest.messagebird.com/messages
 - Port:** 443
 - Max Chars:** 160
- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - MessageBird - Template
- 6) Specify the following in the 'Settings' section:


```
HttpSuccessResponse = "status":"sent"
```
- 7) Specify the following in the 'Text' section:


```
|Authorization|AccessKey PasteYourAccessKeyHere|

|recipients|%RecipientId%
```

|originator|%Sender%|

|body|%Message%|

8) Click Apply

Step 4: Create Recipients tied to the MessageBird method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: MessageBird

Failover: See [Failover Scenarios](#)

ID/PIN: cell phone number, including country code.

<u>Correct</u>	<u>Incorrect</u>
17818290500	781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring API Daze

Step 1: Configure API Daze in PageGate

1) Run PG Admin.

2) Right click Carriers, select Add

3) Specify the following:

Carrier: APIDaze

Protocol: HTTPPOST

Host: api4.apidaze.io/yourAPIkey/sms/send?api_secret=yoursecretAPIcode

NOTE: Paste your API code in to the 'yoursecretAPIcode' section

Port: 443

How to Send Messages

Max Chars: 160

- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - API Daze - Template
- 6) Specify the following in the 'Text' section:
|Content-Type|application/x-www-form-urlencoded|
\013\010
from=YourAPIDazeNumber&to=%RecipientId%&body=%Message%
- 7) Click Apply

Step 4: Create Recipients tied to the API Daze method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:
 - Full Name:** The name of the recipient
 - Recipient:** This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.
 - Type:** Normal
 - Carrier:** API Daze
 - Failover:** See [Failover Scenarios](#)
 - ID/PIN:** 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500
 - Max Chars:** Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

- 3) Click Apply.

Configuring Telnyx

Step 1: Create a Telnyx Account

- 1) Go to <https://www.telnyx.com/>
- 2) Click Sign Up

3) Create your Telnyx Account

Step 2: Create an Access Key for PageGate to use

1) Follow the Telnyx portal setup guide:

<https://developers.telnyx.com/docs/v2/messaging/quickstarts/portal-setup>

2) Follow the Telnyx development environment setup guide to generate your API key:

<https://developers.telnyx.com/docs/v2/messaging/quickstarts/dev-env-setup>

Step 3: Configure Telnyx in PageGate

1) Run PG Admin.

2) Right click Carriers, select Add

3) Specify the following:

Carrier: Telnyx

Protocol: HTTPPOST

Host: api.telnyx.com/v2/messages

Port: 443

Max Chars: 160

4) Click Apply to add the new delivery method.

5) Go to Carriers - Telnyx - Template

6) Specify the following in the 'Settings' section :

HttpPostType = HeadBody

7) Specify the following in the 'Text' section and click Apply:

|To|{%RecipientId%|

|From|A TN authorized to send SMS in your Telnyx Account in international format.

Example: +17818290500|

|Body|{%Message%|

Note: Information on the 'From' parameter can be found in Telnyx's documentation:

<https://developers.telnyx.com/docs/v2/messaging/features/traffic-type>

Step 4: Create Recipients tied to the Telnyx method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

How to Send Messages

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: Twilio

Failover: See [Failover Scenarios](#)

ID/PIN: cell phone number, including country code.

<u>Correct</u>	<u>Incorrect</u>
+17818290500	781829500, 17818290500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring Mobile-Text-Alerts

Step 1: Create a Mobile-Text-Alerts Account

- 1) Go to <https://mobile-text-alerts.com/sms/api>
- 2) Go through the account creation and registration process to receive your API key.

Step 2: Configure Mobile-Text-Alerts in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:

Carrier: Mobile-Text-Alerts

Protocol: HTTPGET

Host: mobile-text-alerts.com/rest/

Port: 80

Max Chars: 160
- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - Mobile-Text-Alerts - Template
- 6) Specify the following in the 'Text' section:

|key|Enter your API Key|

```
|request|send_message|  
|message|%Message%|  
|number|%RecipientId%|
```

7) Click Apply

Step 3: Create Recipients tied to the Mobile-Text-Alerts method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: Mobile-Text-Alerts

Failover: See [Failover Scenarios](#)

ID/PIN: Cell phone number with no spaces or dashes

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

- 3) Click Apply.

Configuring Voximplant

Step 1: Create a Voximplant Account

- 1) Go to <https://voximplant.com/capabilities/sms>
- 2) Click Sign Up
- 3) Create your Voximplant Account.

Step 2: Find your API Access Key and Password for PageGate to use

- 1) Log in to your Voximplant account.
- 2) In the upper right hand corner of the page, click on the dropdown arrow next to your profile icon.

How to Send Messages

- 3) Select API access
- 4) Click 'Show Key' next to the account you want PageGate to use.
- 5) Copy the Key.

Step 3: Configure Voximplant in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:
Carrier: Voximplant
Protocol: HTTPGET
Host: api.voximplant.com/platform_api/A2PSendSms
Port: 443
Max Chars: 6000
- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - Voximplant - Template
- 6) Specify the following in the 'Text' section:
|account_id|Enter Your Account ID Here|
|src_number|Your Voximplant Phone Number|
|dst_numbers|{%RecipientId%}|
|text|{%Message%}|
- 7) Click Apply
- 8) Go to Carriers - Voximplant - Advanced
- 9) Enter your Account ID in the Username field.
- 10) Enter your API Key in to the Password field.
- 11) Click Apply.

Step 4: Create Recipients tied to the Voximplant method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:
Full Name: The name of the recipient

How to Send Messages

3

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: Voximplant

Failover: See [Failover Scenarios](#)

ID/PIN: The 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Public Safety Delivery Services

Some SMS delivery services are designed specifically to work with public safety dispatching to provide mapping, geopositional and other key data to first responders.

To use one of these services, you'll take output from a CAD system and pass various data points like call type, address, crossroads and cross-streets to the service. When the service receives that data, they then put it through a field parser to understand what data exists in which field. As a result, these services usually require data to be passed in a consistently structured order.

Typically speaking, the delivery service will work with you to configure their parser but once the parser is configured, it's important that those field positions remain where they are.

Examples:

- [Active911](#)
- [I Am Responding](#)
- [Chief Messaging](#)
- [CodeMessaging](#)
- [eDispatches](#)
- [FireTextResponse](#)
- [Rover](#)
- [The Fire Horn](#)
- [Who's Responding](#)

How to Send Messages

3

If your SMS delivery service supports multiple APIs, hosts or delivery methods, you can configure them for failover and redundancy. For more information, see the [Failover Scenarios](#) section of the documentation.

For a list of known SMS delivery services, have a look at the [Message Aggregators section of our integration page](#).

Configuring Active911

Active911 supports receiving messages by SNPP and SMTP. As a result, we can configure PageGate to use the better and more direct SNPP method first and fail over to SMTP if something goes wrong.

Special Notes:

- Port 444 will need to be open for SNPP traffic.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure Active911-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:
Carrier: Active911-PRIMARY
Protocol: SNPP
Host: snpp.active911.xcom
Port: 444
Max Chars: 6000
- 4) Go to Carriers - Active911-PRIMARY - Advanced
- 5) Set Priority to: 1
- 6) Click Apply

Step 2: Configure Active911-BACKUP as a delivery method

- 1) Right click Carriers, select Add
- 2) Specify the following:
Carrier: Active911-BACKUP
Protocol: SMTP

How to Send Messages

Host: alert.active911.com

Port: Specify the port required by your SMTP server.

Max Chars: 6000

3) Go to Carriers - Active911-BACKUP - Advanced

4) Specify the following:

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Set 'Priority' to: 1

5) Click Apply

6) Go to Carriers - Active911-BACKUP - Template

7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

8) Click Apply.

Step 3: Configure the failover

1) Go to Carriers - Active911-PRIMARY - Advanced

2) Click the dropdown arrow next to 'Failover' and select: Active911-BACKUP

3) Click Apply

Step 4: Create Recipients tied to the Active911 method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

How to Send Messages

3

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: Active911-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: The Active911 messaging ID. Typically speaking, Active911 will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

For example, if your Active911 ID is 491-x3aAkdbn@alert.active911.com, you would use 491-x3aAkdbn in the ID/PIN field.

Max Chars: 6000

3) Click Apply.

Configuring I Am Responding

I Am Responding supports receiving messages by SMTP/Email and has been investigating the possibility of bringing SNPP services online.

PageGate has the ability to send SMTP/Email messages in one of two ways:

- Direct

Direct delivery configures PageGate to use its own mail server engine to directly negotiate with the outside world.

To configure this method of delivery, you will need to open port 25 and add the public IP address of your PageGate server to the SPF record for your email domain.

- Relayed

Relayed delivery configures PageGate to pass all of its traffic through an SMTP server.

To configure this method of delivery, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Technically, direct delivery is more efficient and reliable because it removes at least one hop in the delivery sequence and also means that PageGate isn't reliant on another mail server for delivery. However, when you use direct delivery for SMTP/Email, you'll need abide by all of the rules required of a mail server because the program will be functioning as a mail server to deliver SMTP/Email on behalf of your email domain.

Due to some of the complications involved in configuring PageGate for direct delivery, it may be more practical to configure PageGate to relay its SMTP/Email traffic through your SMTP/Email server. However, doing so makes PageGate reliant on that mail server for message delivery.

Step 1: Configure IAmResponding as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: IAmResponding

Protocol: SMTP

Host: iaralerts.com

Port: When relaying SMTP, specify the port required by your SMTP server.

When using direct SMTP, set the port to 25.

Max Chars: 6000

- 4) Go to Carriers - IAmResponding - Advanced
- 5) Specify the following:

When relaying SMTP, select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

When using direct SMTP, select 'Deliver directly to destination server'.

How to Send Messages

Set 'Priority' to: 1

- 6) Click Apply
- 7) Go to Carriers - IAmResponding - Template
- 8) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

- 9) Click Apply.

Step 2: Create Recipients tied to the IAmResponding method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: IAmResponding

Failover: See [Failover Scenarios](#)

ID/PIN: The IAR messaging ID. Typically speaking, IAR will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

For example, if your IAR ID is 4912anh@iaralerts.com, you would use 4912anh in the ID/PIN field.

Max Chars: 6000

- 3) Click Apply.

Configuring Chief Messaging

Chief Messaging supports receiving messages by SNPP and SMTP. As a result, we can configure PageGate to use the better and more direct SNPP method first and fail over to SMTP if something goes wrong.

Special Notes:

- Port 444 will need to be open for SNPP traffic.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure ChiefMessaging-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:
Carrier: ChiefMessaging-PRIMARY
Protocol: SNPP
Host: snpp.chiefcadpage.com
Port: 444
Max Chars: 6000
- 4) Go to Carriers - ChiefMessaging-PRIMARY - Advanced
- 5) Set Priority to: 1
- 6) Click Apply

Step 2: Configure ChiefMessaging-BACKUP as a delivery method

- 1) Right click Carriers, select Add
- 2) Specify the following:
Carrier: ChiefMessaging-BACKUP
Protocol: SMTP
Host: alert.active911.com
Port: Specify the port required by your SMTP server.
Max Chars: 6000
- 3) Go to Carriers - ChiefMessaging-BACKUP - Advanced
- 4) Specify the following:
Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

How to Send Messages

3

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Set 'Priority' to: 1

5) Click Apply

6) Go to Carriers - ChiefMessaging-BACKUP - Template

7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

8) Click Apply.

Step 3: Configure the failover

1) Go to Carriers - ChiefMessaging-PRIMARY - Advanced

2) Click the dropdown arrow next to 'Failover' and select: ChiefMessaging-BACKUP

3) Click Apply

Step 4: Create Recipients tied to the Chief Messaging method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: ChiefMessaging-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: The Chief Messaging messaging ID.

Typically speaking, Chief Messaging will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

As an example, if your Chief Messaging address is uxW900@chiefcadpage.com, you would use uxW900 in the ID/PIN field.

Max Chars: 6000

3) Click Apply.

Configuring CodeMessaging

CodeMessaging supports receiving messages by WCTP, SNPP and SMTP. As a result, we can configure PageGate to use the better and more direct WCTP method first, fail over to SNPP if something goes wrong and then have a final last-ditch attempt to deliver it as SMTP/Email.

Special Notes:

- Port 443 will need to be open for WCTP.
- Port 444 will need to be open for SNPP.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure CodeMessaging-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: CodeMessaging-PRIMARY

Protocol: WCTP

Host: www.codepassport.net/codepassport.asp?wci=getWCTP

Port: 443

Max Chars: 1000

- 4) Go to Carriers - CodeMessaging-PRIMARY - Advanced
- 5) Specify the following:

Priority: 1

How to Send Messages

Username: Contact CodeMessaging to obtain this information.

Password: Contact CodeMessaging to obtain this information.

6) Click Apply

Step 2: Configure CodeMessaging-SECONDARY as a delivery method

1) Right click Carriers, select Add

2) Specify the following:

Carrier: CodeMessaging-SECONDARY

Protocol: SNPP

Host: snpp.c-msg.net

Port: 444

Max Chars: 1000

3) Go to Carriers - CodeMessaging-SECONDARY - Advanced

4) Specify the following:

Priority: 1

Password: Enter the CodeMessaging username and password values separated by a single space.

EXAMPLE:

username password

NPSupport PasswordGoesHere

Contact CodeMessaging to obtain your authentication credentials.

5) Click Apply

Step 3: Configure CodeMessaging-FINAL as a delivery method

1) Right click Carriers, select Add

2) Specify the following:

Carrier: CodeMessaging-FINAL

Protocol: SMTP

Host: c-msg.net

Port: Specify the port required by your SMTP server.

Max Chars: 1000

How to Send Messages

3) Go to Carriers - CodeMessaging-FINAL - Advanced

4) Specify the following:

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Set 'Priority' to: 1

5) Click Apply

6) Go to Carriers - CodeMessaging-FINAL - Template

7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

8) Click Apply.

Step 3: Configure the failover system

1) Go to Carriers - CodeMessaging-PRIMARY - Advanced

2) Click the dropdown arrow next to 'Failover' and select: CodeMessaging-BACKUP

3) Click Apply

4) Go to Carriers - CodeMessaging-SECONDARY - Advanced

5) Click the dropdown arrow next to 'Failover' and select: CodeMessaging-FINAL

Step 4: Create Recipients tied to the CodeMessaging method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: Describe the agency being messaged

How to Send Messages

3

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: CodeMessaging-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: The CodeMessaging messaging ID. Typically speaking, CodeMessaging will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

For example, if your CodeMessaging ID is x3aAkdbn@c-msg.net, you would use x3aAkdbn in the ID/PIN field.

Max Chars: 1000

3) Click Apply.

Configuring eDispatches

eDispatches supports receiving messages by SMTP/Email.

PageGate has the ability to send SMTP/Email messages in one of two ways:

- Direct

Direct delivery configures PageGate to use its own mail server engine to directly negotiate with the outside world.

To configure this method of delivery, you will need to open port 25 and add the public IP address of your PageGate server to the SPF record for your email domain.

- Relayed

Relayed delivery configures PageGate to pass all of its traffic through an SMTP server.

To configure this method of delivery, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Technically, direct delivery is more efficient and reliable because it removes at least one hop in the delivery sequence and also means that PageGate isn't reliant on another mail

server for delivery. However, when you use direct delivery for SMTP/Email, you'll need abide by all of the rules required of a mail server because the program will be functioning as a mail server to deliver SMTP/Email on behalf of your email domain.

Due to some of the complications involved in configuring PageGate for direct delivery, it may be more practical to configure PageGate to relay its SMTP/Email traffic through your SMTP/Email server. However, doing so makes PageGate reliant on that mail server for message delivery.

Step 1: Configure eDispatches as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: eDispatches

Protocol: SMTP

Host: edispnoc.com

Port: When relaying SMTP, specify the port required by your SMTP server.

When using direct SMTP, set the port to 25.

Max Chars: 6000

- 4) Go to Carriers - eDispatches - Advanced
- 5) Specify the following:

When relaying SMTP, select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

When using direct SMTP, select 'Deliver directly to destination server'.

Set 'Priority' to: 1

- 6) Click Apply
- 7) Go to Carriers - eDispatches - Template
- 8) Specify the following in the 'Settings' section:

How to Send Messages

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

9) Click Apply.

Step 2: Create Recipients tied to the eDispatches method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: eDispatches

Failover: See [Failover Scenarios](#)

ID/PIN: The eDispatches messaging ID.

Typically speaking, eDispatches will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

For example, if the eDispatches email address is 4912anh@edisnoc.com, you would use 4912anh in the ID/PIN field.

Max Chars: 6000

3) Click Apply.

Configuring FireTextResponse

FireTextResponse supports receiving messages by SMTP/Email.

PageGate has the ability to send SMTP/Email messages in one of two ways:

- Direct

Direct delivery configures PageGate to use its own mail server engine to directly negotiate with the outside world.

To configure this method of delivery, you will need to open port 25 and add the public IP address of your PageGate server to the SPF record for your email domain.

- Relayed

Relayed delivery configures PageGate to pass all of its traffic through an SMTP server.

To configure this method of delivery, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Technically, direct delivery is more efficient and reliable because it removes at least one hop in the delivery sequence and also means that PageGate isn't reliant on another mail server for delivery. However, when you use direct delivery for SMTP/Email, you'll need abide by all of the rules required of a mail server because the program will be functioning as a mail server to deliver SMTP/Email on behalf of your email domain.

Due to some of the complications involved in configuring PageGate for direct delivery, it may be more practical to configure PageGate to relay its SMTP/Email traffic through your SMTP/Email server. However, doing so makes PageGate reliant on that mail server for message delivery.

Step 1: Configure FireTextResponse as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: FireTextResponse

Protocol: SMTP

Host: ftralerts.com or ftralerts.net

Port: When relaying SMTP, specify the port required by your SMTP server.

When using direct SMTP, set the port to 25.

Max Chars: 6000

- 4) Go to Carriers - FireTextResponse - Advanced
- 5) Specify the following:

When relaying SMTP, select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

When using direct SMTP, select 'Deliver directly to destination server'.

Set 'Priority' to: 1

- 6) Click Apply
- 7) Go to Carriers - FireTextResponse - Template
- 8) Specify the following in the 'Settings' section:
From = The value from the username field
EXAMPLE:
From = CAD@notepage.com
- 9) Click Apply.

Step 2: Create Recipients tied to the FireTextResponse method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: FireTextResponse

Failover: See [Failover Scenarios](#)

ID/PIN: The FireTextResponse messaging ID.

Typically speaking, FireTextResponse will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

For example, if the email address is 4912anh@ftralerts.com, you would use 4912anh in the ID/PIN field.

Max Chars: 6000

3) Click Apply.

Configuring Rover

Rover supports receiving messages by WCTP and SMTP. This allows you to configure WCTP as a primary method of delivery with a failover method of SMTP/email.

Special Notes:

- Port 443 will need to be open for WCTP.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure Rover-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: Rover

Protocol: WCTP

Host: websvc.spotteddogtech.com/Rover.WCTP

Port: 443

Max Chars: 6000

- 4) Go to Carriers - Rover - Advanced
- 5) Specify the following:

Priority: 1

- 6) Click Apply

Step 2: Configure Rover-SECONDARY as a delivery method

- 1) Right click Carriers, select Add
- 2) Specify the following:

Carrier: Rover-SECONDARY

Protocol: SMTP

Host: rover.spotteddogtech.com

Port: Specify the port required by your SMTP server.

How to Send Messages

Max Chars: 6000

3) Go to Carriers - Rover-SECONDARY - Advanced

4) Specify the following:

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Set 'Priority' to: 1

5) Click Apply

6) Go to Carriers - Rover-SECONDARY - Template

7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

8) Click Apply.

Step 3: Configure the failover

1) Go to Carriers - Rover-PRIMARY - Advanced

2) Click the dropdown arrow next to 'Failover' and select: Rover-SECONDARY

3) Click Apply

Step 4: Create Recipients tied to the Rover method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

How to Send Messages

3

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: Rover

Failover: See [Failover Scenarios](#)

ID/PIN: Enter the Rover ID. Example: Value@rover.spotteddogtech.com

Max Chars: 6000

3) Click Apply.

Configuring The Fire Horn

The Fire Horn provides a CGI URL for clients to submit messages to and they can use PageGate's HTTPGET protocol to do so.

Step 1: Configure FireHorn as a delivery method

1) Run PG Admin.

2) Right click Carriers, select Add

3) Specify the following and click Apply:

Carrier: Rover

Protocol: HTTPGET

Host: thfirehorn.com/api/software_relay/text_message/YourSiteKeyHere/

NOTE: You will need your Fire Horn site key as part of the host address.

Port: 443

Max Chars: 6000

4) Go to Carriers - FireHorn - Template

5) Specify the following in the 'Text' section:

|message|%Message%|

6) Click Apply

Step 2: Create Recipients tied to the FireHorn method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

How to Send Messages

3

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: FireHorn

Failover: See [Failover Scenarios](#)

ID/PIN: Enter the FireHorn ID

Max Chars: 6000

3) Click Apply.

Configuring Who's Responding

Who's Responding supports receiving messages by SNPP and SMTP. As a result, we can configure PageGate to use the better and more direct SNPP method first and fail over to SMTP if something goes wrong.

Special Notes:

- Port 444 will need to be open for SNPP traffic.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure WhosResponding-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: WhosResponding-PRIMARY

Protocol: SNPP

Host: ds1.whosresponding.com

Port: 444

Max Chars: 6000

- 4) Go to Carriers - WhosResponding-PRIMARY - Advanced
- 5) Set Priority to: 1
- 6) Click Apply

Step 2: Configure WhosResponding-BACKUP as a delivery method

How to Send Messages

1) Right click Carriers, select Add

2) Specify the following:

Carrier: WhosResponding-BACKUP

Protocol: SMTP

Host: ds1.whosresponding.com

Port: Specify the port required by your SMTP server.

Max Chars: 6000

3) Go to Carriers - WhosResponding-BACKUP - Advanced

4) Specify the following:

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Set 'Priority' to: 1

5) Click Apply

6) Go to Carriers - WhosResponding-BACKUP - Template

7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

8) Click Apply.

Step 3: Configure the failover

1) Go to Carriers - WhosResponding-PRIMARY - Advanced

2) Click the dropdown arrow next to 'Failover' and select: WhosResponding-BACKUP

3) Click Apply

Step 4: Create Recipients tied to the WhosResponding method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: Describe the agency being messaged

Recipient: This field is auto-populated and may need to be edited. When integrating with a CAD system, this field will need to reflect CAD's name.

For example, if CAD is going to dispatch 'Palentine Fire District 2' by calling on the name pfd2, that is the value you would use in this field.

Type: Normal

Carrier: WhosResponding-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: The Who's Responding messaging ID.

Typically speaking, Who's Responding will give you an email address per group you want to send to. The prefix of the email address, the part before the @, is what you'll use in the ID/PIN field.

Max Chars: 6000

3) Click Apply.

Medical Industry Delivery Services

Standard SMS/text messaging is not secure, not HIPAA compliant and should never be used to transmit messages with PHI. Some SMS delivery services are designed to work with hospitals, clinics and other medical services to provide HIPAA compliant secure messaging.

In most cases, these services have two methods of operating and some even support both:

- A secure server that hosts all messages

With this method, the SMS service hosts a secure messaging server that they provide you and your users access to. When you send a message, the message is created on the secure server and a notification with login link is sent to the end user's phone. The end user must then log in to the secure server to retrieve their message.

- A secure app on the end-user's device

With this method, the SMS service provides a secure messaging app that is installed on the end user's devices. When you send a message, you initiate the message through the service's secure server and they provide the message to the secure app on the device.

How to Send Messages

3

For more information, have a look at our article on [how secure messaging works](#).

Examples: [TigerConnect](#), [QliqSOFT](#), DocHalo

Configuring TigerConnect

TigerConnect supports receiving messages by WCTP, an HTTPS method of delivery..

Step 1: Configure TigerConnect as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: TigerConnect

Protocol: WCTP

Host: wctp.tigertext.me/wctp

Port: 443

Max Chars: 160

- 4) Go to Carriers - TigerConnect - Advanced

- 5) Specify the following:

Priority: 1

Username: Enter the TigerConnect UserID.

Password: Enter the TigerConnect password.

- 6) Click Apply

Step 2: Create Recipients tied to the TigerConnect method of delivery

- 1) Right click Recipients, select Add.

- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: TigerConnect

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone or pager number

Correct

Incorrect

7818290500

1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring QliqSOFT

QliqSOFT supports receiving messages by POST method to their secure API.

Step 1: Configure QliqSOFT in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:

Carrier: QliqSOFT

Protocol: HTTPPOST

Host: webprod.qliqsoft.com/secure_messages/send

Port: 443

Max Chars: 160

- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - QliqSOFT - Template
- 6) Specify the following in the 'Text' section:

|api_key|your-QliqSOFT-web-API-key|

|to|%RecipientId%|

|from|%Sender%|

|text|%Message%|

NOTE: You will need your QliqSOFT web API key. Information on how to generate that API key can be found in the QliqSOFT documentation:

https://webprod.qliqsoft.com/qliqCONNECT_CloudApiGuide.pdf

- 7) Click Apply

Step 4: Create Recipients tied to the QliqSOFT method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

How to Send Messages

Type: Normal

Carrier: QliqSOFT

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring DocHalo

DocHalo supports receiving messages by WCTP, an HTTPS method of delivery..

Step 1: Configure DocHalo as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: DocHalo

Protocol: WCTP

Host: wctp.halospectrum.com/message

Port: 443

Max Chars: 160

- 4) Go to Carriers - DocHalo - Advanced
- 5) Specify the following:

Priority: 1

Username: Enter the DocHalo UserID.

Password: Enter the DocHalo password.

- 6) Click Apply

Step 2: Create Recipients tied to the DocHalo method of delivery

- 1) Right click Recipients, select Add.
 - 2) Specify the following:
- Full Name:** The name of the recipient

How to Send Messages

3

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: DocHalo

Failover: See [Failover Scenarios](#)

ID/PIN: The 8 or 12 digit DocHalo HDID
Examples: DH123456 or DH0001020A3B

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring Voalte

Voalte supports receiving messages by WCTP, an HTTPS method of delivery..

Step 1: Configure Voalte as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: Voalte

Protocol: WCTP

Host: cws01.voalte.net/wctp

Port: 50100

Max Chars: 6000

- 4) Go to Carriers - Voalte - Advanced
- 5) Specify the following:
Priority: 1
- 6) Click Apply
- 7) Go to Carriers - Voalte - Template
- 8) Specify the following in the 'Settings' section:
StripNonPrintable = True
From = Your whitelisted Voalte Sender ID
- 9) Click Apply.

Step 2: Create Recipients tied to the Voalte method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: Voalte

Failover: See [Failover Scenarios](#)

ID/PIN: Voalte Contact ID

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring Zinc

ServiceMax's Zinc supports receiving messages by POST method to their secure API.

For this configuration, you will need to know the Message Bot Identification Token and information on both using their API and determining the token can be found in the Zinc documentation: <https://community.servicemax.com/t5/Zinc-Help-Articles/Zinc-How-to-Use-Messaging-API/ba-p/17738>

Step 1: Configure Zinc in PageGate

1) Run PG Admin.

2) Right click Carriers, select Add

3) Specify the following:

Carrier: Zinc

Protocol: HTTPPOST

Host: api.zincit.io/hooks/v1/:id/message

NOTE: The :id portion of the URL is determined by the Message Bot Identification Token. Please see [Zinc's documentation](#) for more information.

Port: 443

Max Chars: 6000

4) Click Apply to add the new delivery method.

5) Go to Carriers - Zinc - Template

How to Send Messages

6) Specify the following in the 'Text' section:

```
|api_key|your-Zinc-web-API-key|
|to|%RecipientId%|
|from|%Sender%|
|text|%Message%|
```

NOTE: You will need your Zinc web API key. Please see [Zinc's documentation](#) for more information.

7) Click Apply

Step 4: Create Recipients tied to the Zinc method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: QliqSOFT

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring Spok Mobile

Spok Mobile supports receiving messages by WCTP, an HTTPS method of delivery.

Step 1: Configure SpokMobile as a delivery method

1) Run PG Admin.

2) Right click Carriers, select Add

3) Specify the following and click Apply:

Carrier: SpokMobile

Protocol: WCTP

Host: wctp.usamobility.net/wctp

Port: 443

Max Chars: 160

Step 2: Create Recipients tied to the SpokMobile method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: SpokMobile

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone or pager number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Carrier Enterprise Systems

The major North American carriers each have their own respective enterprise messaging platforms and these gateways are ultimately faster and more reliable than the free, public SMTP-to-SMS system. These platforms also typically provide multiple APIs, allowing PageGate to use all of them in a failover capacity.

Access to these gateways is usually not provided free of charge and will require a contract of some variety with the carrier.

- [Verizon Enterprise Messaging Access Gateway \(EMAG\)](#)
- [AT&T FirstNet](#)
- [Sprint Enterprise Messaging Gateway \(EMG\)](#)

Configuring EMAG

To register on [Verizon's EMAG](#) platform, you'll need to contact your Verizon representative, who will provide an intake form.

In the registration process, you'll need to specify whether your EMAG account has the ability to send to all cell phones, also called inter-carrier operability, or just Verizon cell phones.

Note for Public Safety: Verizon offers free access to dispatch related public safety organizations if you ONLY message Verizon phones through EMAG. Enabling inter-carrier operability requires a paid contract.

On the intake form, you're interested in the WCTP, SNPP and SMTP APIs. With all three, we can configure a failover system where the program will use the best method of delivery (WCTP) first, then fail over to the second best (SNPP) and if that fails, it makes a last ditch attempt to send it out via email (SMTP).

For the WCTP and SNPP APIs, you'll need to provide the public IP address your PageGate server resides behind. You'll also be asked to create a username and password that are unique to each API and we'll need those credentials for PageGate's configuration.

For the SMTP API, you'll need to specify whether you have a cloud based solution, like Office 365, or a traditional SMTP server and provide your email domain and server IP information.

It's important to note that EMAG ties your API credentials to your public IP address. If your public IP address changes, you'll definitely want to update that with EMAG and, if possible, communicate the shift ahead of time

Special Notes:

- Port 443 will need to be open for WCTP.
- Port 444 will need to be open for SNPP.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure EMAG-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following and click Apply:

Carrier: EMAG-PRIMARY

Protocol: WCTP

How to Send Messages

Host: wctp.emag.vzw.com/wctp/wctp

Port: 443

Max Chars: 160

4) Go to Carriers - EMAG-PRIMARY - Advanced

5) Specify the following:

Priority: See [Prioritizing Messages](#)

Username: The EMAG WCTP API Username

Password: The EMAG WCTP API Password

6) Click Apply

7) Go to Carriers - EMAG-PRIMARY - Template

8) Specify the following in the 'Settings' section:

StripNonPrintable = True

9) Click Apply.

Step 2: Configure EMAG-SECONDARY as a delivery method

1) Right click Carriers, select Add

2) Specify the following:

Carrier: EMAG-SECONDARY

Protocol: SNPP

Host: snpp.emag.vzw.com

Port: 444

Max Chars: 160

3) Go to Carriers - EMAG-SECONDARY - Advanced

4) Specify the following:

Priority: See [Prioritizing Messages](#)

Password: Enter the EMAG SNPP username and password values separated by a single space.

EXAMPLE:

username password

NPSupport PasswordGoesHere

5) Click Apply

Step 3: Configure EMAG-FINAL as a delivery method

How to Send Messages

1) Right click Carriers, select Add

2) Specify the following:

Carrier: EMAG-FINAL

Protocol: SMTP

Host: vzvmg.biz

Port: Specify the port required by your SMTP server.

Max Chars: 160

3) Go to Carriers - EMAG-FINAL - Advanced

4) Specify the following:

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Priority: See [Prioritizing Messages](#)

5) Click Apply

6) Go to Carriers - EMAG-FINAL - Template

7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

8) Click Apply.

Step 4: Configure the failover system

1) Go to Carriers - EMAG-PRIMARY - Advanced

2) Click the dropdown arrow next to 'Failover' and select: EMAG-SECONDARY

3) Click Apply

4) Go to Carriers - EMAG-SECONDARY - Advanced

5) Click the dropdown arrow next to 'Failover' and select: EMAG-FINAL

6) Click Apply

Step 4: Create Recipients tied to the EMAG-PRIMARY method of delivery

1) Right click Recipients, select Add.

2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: EMAG-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

Configuring FirstNet

To begin the process of registering on FirstNet, you'll want to contact your AT&T representative or go to <https://www.firstnet.com/signup>

During this registration process, you'll need to let your Firstnet representative know that you'll need access to the Firstnet Messaging Portal (<https://messaging.firstnet.com/>).

FirstNet supports receiving messages by SNPP and SMTP. With all both, we can configure a failover system where the program will use the best method of delivery (SNPP) first, then fail over to the second best (SMTP).

To use the SNPP gateways, you will need login access to the AT&T FirstNet Admin or Business Notification Center web portals.

Once there, on the left hand side, go to Administration and you should see a 'Generate Credentials' option. You will need both the username and the password for PageGate.

Special Notes:

- Port 444 will need to be open for SNPP.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure FirstNet-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:
 - Carrier:** FirstNet-PRIMARY
 - Protocol:** SNPP
 - Host:** snpp.firstnet.com
 - Port:** 444
 - Max Chars:** 999
- 4) Click Apply.
- 5) Go to Carriers - FirstNet-PRIMARY - Advanced
- 6) Enter your FirstNet SNPP API username and password in to the Username and Password fields
 - Username:** Your FirstNet API Username

 - Password:** Your FirstNet API Password
- 7) Click Apply.

Step 2: Configure FirstNet-FINAL as a delivery method

- 1) Right click Carriers, select Add
- 2) Specify the following:
 - Carrier:** FirstNet-FINAL
 - Protocol:** SMTP
 - Carrier Domain:** 1waysms.firstnet.com
 - Port:** Specify the port required by your SMTP server.
 - Max Chars:** 1000
- 3) Go to Carriers - FirstNet-FINAL - Advanced
- 4) Specify the following:

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

How to Send Messages

3

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Priority: See [Prioritizing Messages](#)

- 5) Click Apply
- 6) Go to Carriers - FirstNet-FINAL - Template
- 7) Specify the following in the 'Settings' section:

From = The value from the username field

EXAMPLE:

From = CAD@notepage.com

- 8) Click Apply.

Step 4: Configure the failover system

- 1) Go to Carriers - FirstNet-PRIMARY - Advanced
- 2) Click the dropdown arrow next to 'Failover' and select: FirstNet-FINAL
- 3) Click Apply

Step 4: Create Recipients tied to the FirstNet-PRIMARY method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: FirstNet-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone number

<u>Correct</u>	<u>Incorrect</u>
7818290500	1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

How to Send Messages

3

3) Click Apply.

Configuring EMG

To begin the process of registering on Sprint's EMG, you'll want to contact your Sprint representative or go to <https://emg.sprint.com>

EMG supports receiving messages by SNPP and SMTP. With all both, we can configure a failover system where the program will use the best method of delivery (SNPP) first and if that fails, it makes a last ditch attempt to send it out via email (SMTP).

Special Notes:

- Port 444 will need to be open for SNPP.
- To configure the SMTP failover, you will need to know your SMTP server address, port to connect on and required authentication credentials.

Step 1: Configure Sprint-PRIMARY as a delivery method

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:

Carrier: Sprint-PRIMARY

Protocol: SNPP

Host: snpp.messaging.sprint.com

Port: 444

Max Chars: 160

- 4) Click Apply.

Step 2: Configure Sprint-FINAL as a delivery method

- 1) Right click Carriers, select Add
- 2) Specify the following:

Carrier: Sprint-FINAL

Protocol: SMTP

Host: messaging.sprintpcs.com

Port: Specify the port required by your SMTP server.

Max Chars: 160
- 3) Go to Carriers - Sprint-FINAL - Advanced
- 4) Specify the following:

How to Send Messages

3

Select the 'Deliver through outgoing mail server' option and enter your SMTP server address in the 'Relay Server' field.

For example, if you use Office365, you would use smtp.office365.com as the relay server address.

Specify the required username and password.

Priority: See [Prioritizing Messages](#)

- 5) Click Apply
- 6) Go to Carriers - Sprint-FINAL - Template
- 7) Specify the following in the 'Settings' section:
From = The value from the username field
EXAMPLE:
From = CAD@notepage.com
- 8) Click Apply.

Step 3: Configure the failover system

- 1) Go to Carriers - Sprint-PRIMARY - Advanced
- 2) Click the dropdown arrow next to 'Failover' and select: Sprint-FINAL
- 3) Click Apply

Step 4: Create Recipients tied to the Sprint-PRIMARY method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: Sprint-PRIMARY

Failover: See [Failover Scenarios](#)

ID/PIN: 10 digit cell phone number

Correct

Incorrect

How to Send Messages

3

7818290500

1781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

3) Click Apply.

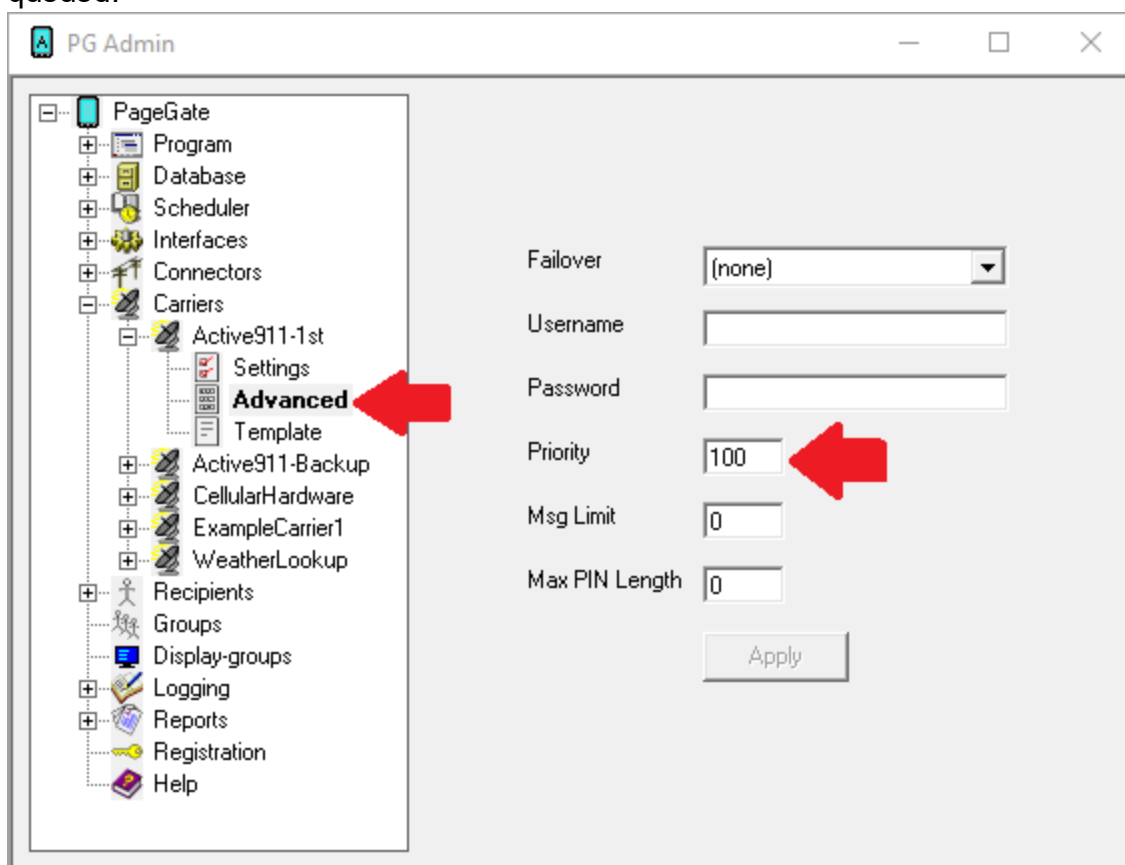
Prioritizing Messages

PageGate allows you to prioritize your traffic by setting a priority value at both a carrier and recipient level and the 'Priority' field can be found in the Advanced sub-section of every carrier and recipient.

The scale of priority is 1 (the most important) to 32000 (the least important). By default, all carriers and recipients start with a priority value of 100.

Entries that share a priority value are treated equally. Entries with lower values are treated as more important, entries with higher values are treated as less important.

Carrier level priority determines the order in which carriers should have their messages queued.

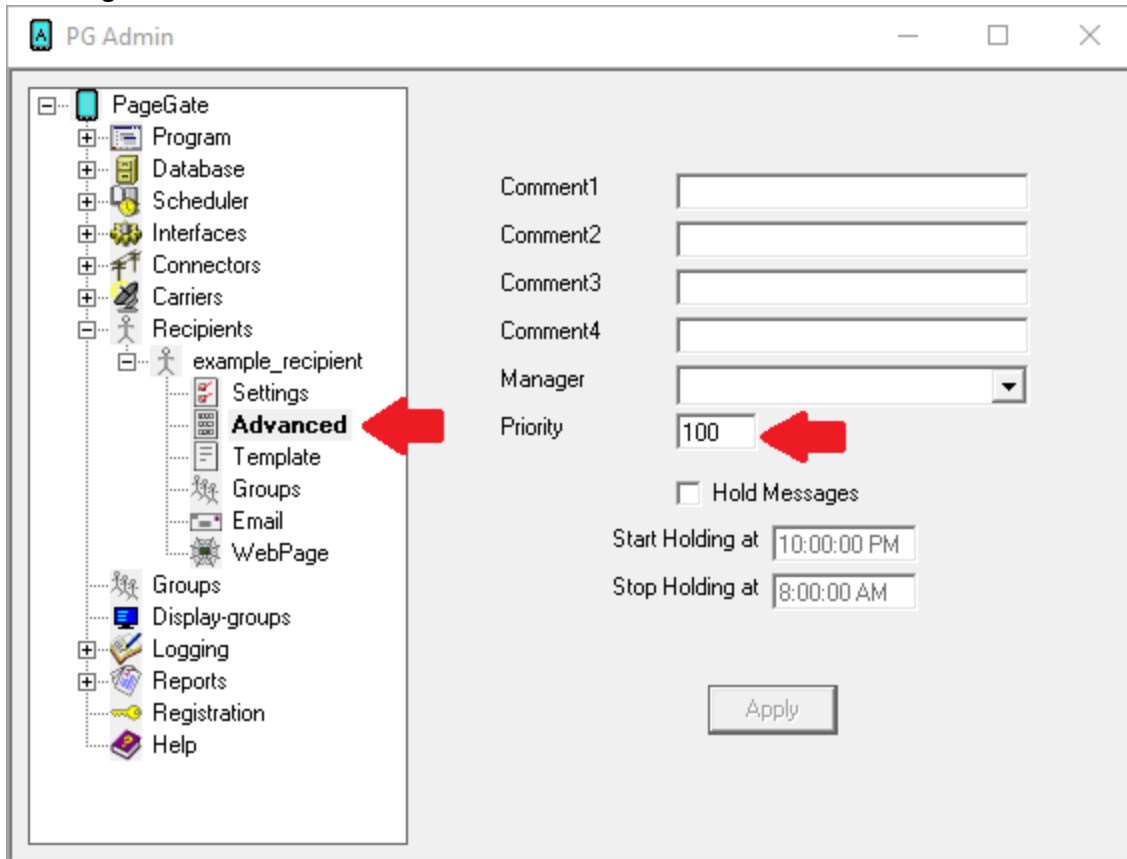


For example, let's say that you use I Am Responding and Active911's message aggregation service and that you also deliver regular email, SMS and pages through PageGate. If you set the priority value for I Am Responding and Active911 to something

lower than the priority value for the other carriers, PageGate will always schedule the I Am Responding and Active911 messages to go out before anything else.

Carrier priority values are determined in a carrier's [Advanced sub-section](#).

Recipient level priority determines the order in which recipients should receive their messages.



For example, let's say that you had a list of 100 recipients and that we had 10 recipients who needed to receive their traffic before anyone else. If you set the priority value for those recipients to something lower than the priority value for the other recipients, PageGate will always schedule their messages first. To give a slightly more practical example, let's say that you have a dispatch supervisor or fire marshal or IT director that needed to be notified before anyone else in the system. You could edit the value of that recipient's Priority field to be lower than the other recipients and that would ensure that they always received their messages first.

Recipient priority values are determined in a recipient's [Advanced sub-section](#).

How to send Pages

PageGate can be configured to deliver messages to pagers with modem, serial or internet methods of delivery. Pagers receive their service through a local paging transmitter or from carriers like USA Mobility and American Messaging and this determines what you'll need to configure in PageGate.

On-site paging transmitters, like those made by Zetron, Glenayre and Motorola, support receiving messages via modem or serial line by the TAP protocol and PageGate supports TAP delivery by either method. For more information on how to configure either a modem or serial TAP carrier, have a look at the [Creating a TAP Carrier](#) section of the documentation.

Pagers hosted through a paging company can be messaged with any protocols the carrier supports.

For example, American Messaging and USA Mobility support the SNPP, WCTP, SMTP and TAP protocols. SNPP and WCTP are more efficient and reliable than SMTP and should be implemented where possible. If you're looking for a method of delivery that isn't reliant on your internet connection, most pager companies also still support dial-in TAP terminals.

PageGate also supports sending TAP by TCP/IP as well.

[Click here](#) for a list of known SNPP hosts.

[Click here](#) for steps to create SNPP carriers.

[Click here](#) for a list of known WCTP hosts.

[Click here](#) for steps to create WCTP carriers.

[Click here](#) for a list of known SMTP addresses.

[Click here](#) for steps to create SMTP carries.

[Click here](#) for a list of known TAP terminals.

[Click here](#) for steps to create TAP carriers.

How to send Email

SMTP is the protocol used to send email of all varieties, including those delivered as SMS; nearly every US and Canadian carrier supports text messaging to their phones using a public access SMTP/Email gateway.

PageGate has a full mail server engine as a core part of the program and this gives you the ability to send SMTP/Email messages in one of two ways: Direct and Relayed

To explain the difference between the two, direct delivery allows PageGate to use its own mail server engine to directly negotiate with the receiving mail servers. Relayed delivery configures PageGate to pass all of its traffic through a verified SMTP server for delivery, like an email client.

Technically, direct delivery is more efficient and reliable because it removes at least one hop in the delivery sequence and also means that PageGate isn't reliant on another mail server for delivery. However, when you have PageGate use direct delivery for SMTP/Email, it has to abide by all of the rules required of a mail server because the program will be functioning as a mail server to deliver SMTP/Email on behalf of your email domain. This is incredibly important to note because, in an effort to combat spam, all major US and Canadian carriers pay respect to SPF records.

SPF stands for "Sender Policy Framework" and is a record of public IP addresses that are allowed to deliver email on behalf of your email domain. This record is published by whoever owns and operates the email domain. If the PageGate server's public IP address is not part of your email domain's SPF record, the major carriers will filter your messages as potential spam for failing the SPF validation check. If you continue to send traffic that fails SPF validation checks, the receiving servers will eventually blacklist your site from sending any traffic to them.

To check to see if port 25 is open, run the [Port Scan](#) in the Program - Tools section of the PageGate Admin.

To verify that your site can perform the necessary MX lookups and to discover what your SPF record is, run [DNS Lookup](#) in the Program - Tools section of the PageGate admin.

Due to some of the complications involved in configuring PageGate for direct delivery, it is usually more practical to configure PageGate to relay its SMTP/Email traffic through your company or ISP's SMTP/Email server. However, doing so makes PageGate reliant on that mail server for message delivery.

For information on how to configure a "Relayed Delivery" SMTP carrier, [click here](#).

For information on how to configure a "Direct Delivery" SMTP carrier, [click here](#).

Using webforms, Web APIs and REST APIs

PageGate supports the ability to use HTTP GET and POST methods which can be used to submit web forms, request information from websites and/or WEB APIs and interact with REST APIs.

The website, web form or API you interact with will directly determine what you need to configure in PageGate.

GET and POST are conceptually similar to each other but perform their functions in completely different ways. The GET method carries parameters in the URL string and the POST method carries parameters in the request body. GET can be used to interact with WEB APIs and POST is used to interact with webforms and REST APIs and both can be used to request information from and submit information to them.

- [HTTPGET](#)

As an example, ViaNett (vianett.com) is an SMS delivery service in Europe that provides a Web API to send SMS. This is a sample HTTP GET request to send a message via their Web API:

```
https://smsc.vianett.no/v3/send?  
username=UsernameValue&password=PasswordValue&msgid=1234&tel=47111111111  
&msg>Hello+World&pricegroup=300&campaignid=12345
```

As you can see, all variables are declared as part of the URL. If you need to reference JSON, XML or something similar, you'll want to use the POST method.

- [HTTPPOST](#)

Web APIs, REST APIs and webforms all have different requirements for the headers and contents they need to see.

Web forms you interact with, whether it be PHP, CGI or something else, will have certain parameters that determine whether you're requesting or submitting data and you'll need to know each required parameter and the value that should be passed in it.

Some APIs want the body in JSON format, others want XML, others will accept both and leave it to the end user to determine which they want to use.

It's important to gather as much information as you can about what API or webform you're interacting with and their documentation will typically provide required variables and how they need to be implemented.

Templates for HTTP GET Carriers

Templates for HTTP GET carriers tell PageGate what variables and data to declare in the URL passed to the API.

All variables are declared in the 'Text' section of the template as follows:

```
|<name1>|<value1>|  
|<name2>|<value2>|  
etc.
```

Each value you specify will need to correspond to a variable that the receiving API will recognize. For example, let's say we wanted to send a message to an API that looked for to, from, subject and text as variables. We would need to format the text template as follows:

```
|to|%RecipientId%|  
|from|%Sender%|  
|subject|%Subject%|  
|text|%Message%|
```

NOTE: | a pipe character.

The following parameters can be specified in the 'Settings' section of an HTTP GET carrier

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when sending and receiving SMS through this carrier. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This parameter overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
HttpErrorResponse	Command String See Description	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a GET requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p>

		<p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word failed: <code>HttpErrorResponse = failed</code></p> <p>This would scan for the word failed in the <code><Reply></code> tag: <code>HttpErrorResponse = <Reply>failed</Reply></code></p> <p>This would scan for the word reject or the word failed in the <code><Reply></code> tag: <code>HttpErrorResponse = reject <Reply>failed</Reply></code></p> <p>This would scan for the word failed OR the word reject OR the phrase object not found: <code>HttpErrorResponse = failed reject object not found</code></p> <p>Any response from the web server that includes the search term(s) will be treated as a failure.</p>
HttpProtocol	HTTP S HTTP	<p>Example: <code>HttpProtocol = HTTPS</code></p> <p>This parameter forces the use of HTTPS regardless of the port used to connect.</p>
HttpResponseTimeout	Millise conds	<p>Example: <code>HttpResponseTimeout = 1000</code></p> <p>This variable forces PageGate to wait for the specified number of milliseconds (1000 = 1 second) for a response to the HTTP communication request.</p>
HttpSuccessResponse	Com mand String See Descr iption	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a GET requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p>

		<p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word success: <code>HttpSuccessResponse = success</code></p> <p>This would scan for the word Acknowledged in the <Reply> tag: <code>HttpSuccessResponse = <Reply>Acknowledged</Reply></code></p> <p>This would scan for the word hijinks or the word Acknowledged in the <Reply> tag: <code>HttpSuccessResponse =hijinks <Reply>Acknowledged</Reply></code></p> <p>This would scan for the word success OR the word hijinks OR the word actualization: <code>HttpSuccessResponse = success hijinks actualization</code></p> <p>Any response from the web server that includes the search term(s) will be treated as a success.</p>
InterMessageDelay	Milliseconds	<p>Example: <code>InterMessageDelay = 1000</code></p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered. It could be possible for PageGate to send a large number of data requests too quickly and this prevents that from happening.</p>
ParameterDelimiter	See Description	<p>Example: <code>ParameterDelimiter = !</code></p> <p>Message templates for HTTP GET carriers usually involve parameter + variable pairs that are declared in this format: <code> parameter variable </code></p> <p>ParameterDelimiter allows you to change the 'wrapping' character from a pipe () to something else, like an exclamation mark (!).</p>
ResponseIncludeRequest	True False	<p>Example: <code>ResponseIncludeRequest = True</code></p> <p>Setting this parameter to true tells PageGate to include the original request sent to the API when recording responses.</p>
ResponsePath	File Path	<p>Example: <code>ResponsePath = c:\PageGateData\APIResponses\</code></p> <p>This parameter allows you to specify the folder that responses should be written to. If API responses do not need to be tracked, this parameter is not required.</p>

ResponseRecipient	Recipient Group	<p>Example: ResponseRecipient = groupname</p> <p>When the HTTP GET or POST transaction completes, the specified recipient or group will receive any responses from the receiving API/web host. For more information, please see the Two Way Messaging section of the documentation.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>
StripCharacters	See description	<p>Example: StripCharacters = ~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.</p>
To	Text	<p>To = 7818290500</p> <p>This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Templates for HTTP POST Carriers

Templates for HTTP POST carriers tell PageGate how to interact and how to structure data passed to a web form, Web API or REST API.

It's important to gather as much information as you can about what API or webform you're interacting with and their documentation will typically provide required variables and how they need to be implemented.

Conceptually, the 'Settings' section of a POST carrier template tells PageGate if the API requires any special interaction and the 'Text' section determines what variables to declare and how to structure the message body.

These values can be used in the 'Settings' section of an HTTP POST carrier:

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when sending and receiving SMS through this carrier. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This parameter overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
HttpErrorResponse	Command String See Description	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a POST requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p>

		<p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word failed: <code>HttpErrorResponse = failed</code></p> <p>This would scan for the word failed in the <code><Reply></code> tag: <code>HttpErrorResponse = <Reply>failed</Reply></code></p> <p>This would scan for the word reject or the word failed in the <code><Reply></code> tag: <code>HttpErrorResponse = reject <Reply>failed</Reply></code></p> <p>This would scan for the word failed OR the word reject OR the phrase object not found: <code>HttpErrorResponse = failed reject object not found</code></p> <p>Any response from the web server that includes the search term(s) will be treated as a failure.</p>
HttpPostType	<p>HeadBody</p> <p>UrlHeadBody</p>	<p>HttpPostType = HeadBody configures an HTTP POST carrier to provide header information followed by structured data, usually JSON or XML, in the body.</p> <p>HttpPostType = UrlHeadBody configures an HTTP POST carrier to pass parameters in the URL, more like the POST method. This is not a typical use of HTTP POST but is used/required by some APIs.</p>
HttpProtocol	<p>HTTPS</p> <p>HTTP</p>	<p>Example: <code>HttpProtocol = HTTPS</code></p> <p>This parameter forces the use of HTTPS or HTTP regardless of the port used to connect.</p>
HttpResponseTimeout	<p>Milliseconds</p>	<p>Example: <code>HttpResponseTimeout = 1000</code></p> <p>This variable forces PageGate to wait for the specified number of milliseconds (1000 = 1 second) for a response to the HTTP communication request.</p>
HttpSuccessResponse	<p>Command String</p>	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p>

	See Description	<p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a POST requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word success: <code>HttpSuccessResponse = success</code></p> <p>This would scan for the word Acknowledged in the <Reply> tag: <code>HttpSuccessResponse = <Reply>Acknowledged</Reply></code></p> <p>This would scan for the word hijinks or the word Acknowledged in the <Reply> tag: <code>HttpSuccessResponse =hijinks <Reply>Acknowledged</Reply></code></p> <p>This would scan for the word success OR the word hijinks OR the word actualization: <code>HttpSuccessResponse = success hijinks actualization</code></p> <p>Any response from the web server that includes the search term(s) will be treated as a success.</p>
InterMessageDelay	Millisecons	<p>Example: <code>InterMessageDelay = 1000</code></p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered. It could be possible for PageGate to send a large number of data requests too quickly and this prevents that from happening.</p>
ParameterDelimiter	See Description	<p>Example: <code>ParameterDelimiter = !</code></p> <p>Message templates for HTTP POST carriers usually involve parameter + variable pairs that are declared in this format: <code> parameter variable </code></p> <p>ParameterDelimiter allows you to change the 'wrapping' character from a pipe () to something else, like an exclamation mark (!).</p>
ResponseIncludeRequest	True False	<p>Example: <code>ResponseIncludeRequest = True</code></p>

		Setting this parameter to true tells PageGate to include the original request sent to the API when recording responses.
Response Path	File Path	Example: ResponsePath = c:\PageGateData\APIResponses\ This parameter allows you to specify the folder that responses should be written to. If API responses do not need to be tracked, this parameter is not required.
Response Recipient	Recipient Group	Example: ResponseRecipient = groupname When the HTTP POST transaction completes, the specified recipient or group will receive any responses from the receiving API/web host. For more information, please see the Two Way Messaging section of the documentation.
StripAlpha	True False	Example: StripAlpha = True Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'
StripCharacters	See description	Example: StripCharacters = \~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True False	Example: StripNonPrintable = True Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
To	Text	Example: To = 7818290500 This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True

		Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.
--	--	------------------------------------------------------------------------------------------------------------------------------

This is how to use the 'Text' section of an HTTP POST carrier:

Each value you specify will need to correspond to a variable that the receiving API will recognize. For example, let's say we wanted to send a message using [Twilio's API](#). We know they're looking that looked for to, from and body as variables. We would need to format the text template as follows:

```
|to|%RecipientId%|  
|from|%Sender%|  
|body|%Message%|
```

NOTE: | is a pipe character and is found on most keyboards above the backslash character, \.

Some APIs, like [MessageBird](#), require an authentication variable in the body of the POST and those are declared like any other variable.

Example:

```
|Authorization|AccessKey PasteYourAccessKeyHere|
```

```
|recipients|%RecipientId%|  
|originator|%Sender%|  
|body|%Message%|
```

Miscellaneous Methods

Outside of SMS, email and paging, PageGate has the ability to communicate in other methods to a variety of sources and devices. PageGate's miscellaneous delivery methods are:

- [XMPP](#) (Extensible Messaging and Presence Protocol)

This protocol allows PageGate to communicate with an XMPP server/chat system such as Jabber. [Click here](#) to access the steps to create an XMPP carrier.

- [FAX](#)

This protocol allows PageGate to use a faxmodem to deliver messages to destination Fax machines. [Click here](#) to access the steps to create a FAX carrier.

- [FAXSrv](#)

This protocol allows PageGate to pass facsimile messages to either a local or network available Microsoft fax server. [Click here](#) to access the steps to create a FAXSrv carrier.

- [RawSerial](#)

This protocol allows PageGate to deliver messages as raw data strings over a serial port. [Click here](#) to access the steps to create a RawSerial carrier.

- [RawTCP](#)

This protocol allows PageGate to deliver messages by Raw TCP to a host IP. [Click here](#) to access the steps to create a RawTCP carrier.

- [File](#)

This protocol allows PageGate to output a message as an ascii formatted file to a specified directory. This is incredibly useful when implementing the PageGate Filter Pack as it allows you to put messages through for secondary processing. This protocol also, just as an example, allows you to keep a hard record of messages that are sent from the program by keeping a copy of every message sent as a file in the specified directory. [Click here](#) to access the steps to create a File carrier.

- [TONE/Touch Tone/DTMF](#) (Dual Tone Multi-Frequency)

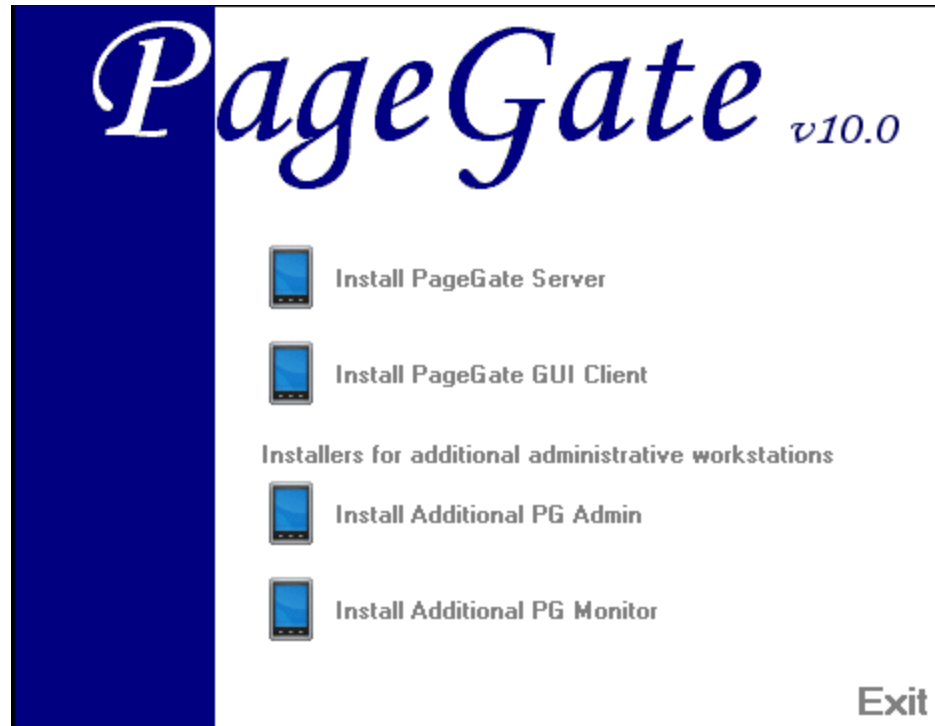
This protocol allows PageGate to use a dial-up modem to deliver touch tones to older numeric-only pagers and similar equipment. [Click here](#) to access the steps to create a TONE carrier.

- [UCP](#) (Universal Computer Protocol)

A delivery protocol for analog modem, or direct serial communications to a wireless provider. [Click here](#) to access the steps to create a UCP carrier.

PageGate is designed to be a highly modular application that can send messages to cell phones, pagers, email addresses or any combination of the three. This design also allows PageGate's modules to be run across multiple computers and platforms. The following installation information covers installation of PageGate on a single computer.

When you run PageGate's installation file, you'll be prompted to perform one of four installations:



The Install PageGate Server option should only be run on the system that will be hosting the database and functioning as the PageGate server. If you want to grant a workstation access to the PageGate Admin, you should use the "Install Additional PG Admin" option.

The PageGate Server installation will install the following components:

- **PageGate Server**

This is the control module for PageGate. It is constantly running in the background whenever the PageGate application is running. It monitors, starts, and stops the other PageGate modules.

- **PageGate Scheduler**

This module processes all messages sent by PageGate. It manages scheduled and repeating messages, in addition to scheduling the delivery of each message by a Connector.

- **PageGate Connectors**

The Connectors are responsible for the actual delivery of the messages.

- **PageGate Interfaces**

These modules (GetAscii, GetMail, GetWeb, GetTAP, GetSerial and GUI) accept messages from various sources, into the PageGate application.

- **PageGate Add-ons**

Additional server processes than can be run to accept messages into the PageGate Server: SNPP Server Add-on, Touch-Tone Receiver Add-on (require the GetAscii interface for operation)

The PageGate GUI Client installation will install the following components:

- **PageGate Client**

A Windows based program installed on workstations on the same network as the PageGate server. This program provides a graphical user interface to send and schedule messages.

The Additional PG Admin installation will install the following components:

- **PageGate Admin**

The administration program is used to setup, control, and maintain the PageGate application.

The Additional PG Monitor installation will install the following components:

- **PageGate Monitor**

The monitoring application allows for real-time monitoring of the PageGate application from the local machine, or from any other workstation that is connected to the PageGate server via TCP/IP.

PageGate Server

The PageGate Server is the main processing portion of PageGate. All messages, modules, and interfaces are contained within and run by the PageGate Server. This installation should only be performed on the system intended to host the PageGate Server and its data.

Minimum system requirements:

Operating System: Windows 7, Windows 8, Windows 10, Server 2012, Server 2016,

Server 2019, Server 2022

Processor: 1.0 Ghz Processor or better.

Memory: 1GB RAM

Storage: 100 MB of free disk space (for application and database)

Communications: PageGate supports internet, hardware and TCP based communications. Please review the [How PageGate Sends Messages](#) section of the documentation to understand what is required to deliver your traffic.

Please reference the [Installation Steps](#) section of the manual for information on how to install the program.

Installation Steps

- [Downloaded installation file](#)

The classic installation method. Download an executable, run it and go through the installation steps.

- [Promptless Automatic installation](#)

A more advanced installation method for deploying the PageGate server via command line parameters passed to the installation.

- [Upgrading from a previous version](#)

Have a previous version? These steps will guide you through the process to upgrade your existing installation to version 10.

Installation by Download

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>
- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, you want to save the file.
- 5) Save the pgsetup.exe file somewhere easy to reference, like the Desktop.
- 6) After the file has been downloaded, right click on it and select "Run As Administrator".
- 7) When prompted, select the Install PageGate Server option.
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click on Next.

- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click on Next.
- 11) In the next step, you will be prompted for the PageGate Server installation directory. By default, the installation directory is C:\Program Files(x86)\NotePage\PageGate\
The installation directory can be changed to any local path. Click Next to proceed.
- 12) In the next step, changes to the program group PageGate will appear under in the start menu can be made. Click Next to proceed.
- 13) Select whether you want to create a Desktop Icon for PageGate and click on Next.
- 14) Click Install to begin installing the PageGate Server.
- 15) Click Finish to finalize the installation.

Be sure to make a copy of the installation file and store it somewhere safe. You will need it again if you ever need to reinstall PageGate, or want to install the Additional PageGate Administrator, Additional PageGate Monitor, or GUI Client on a workstation.

Promptless Automatic Installation

Any part of PageGate can be installed unattended. This helps network administrators to automate the program's installation.

- 1) Start the PageGate setup program as usual ([see above](#)).
- 2) When the list of installation options screen is shown, exit the installation.
- 3) Browse in to the c:\Users\<<your_user>\AppData\Local\Temp\ directory.
- 4) Copy the 'PGSetup' folder to another location.
- 5) Use the following command line syntax to run a promptless install.

PageGate Server Installation File: pgssetup.exe

Command line options:

/silent	Installs without prompts. Does display progress screens, will prompt for reboot if necessary.
/verysilent	Installs without prompts, display screens and will reboot without prompting, if necessary.
/norestart	This flag prevents the installation from rebooting, even if necessary.
/dir ="x:\dirname"	This flag overrides the default installation directory.

/group="folder name"	This flag overrides the default Windows Start Menu group name.
/noicons	This flag will prevent the installation from creating the desktop icons for the program.

The <path> option should be replaced with the path to where you copied the 'PGSetup' folder in step #2 above.

Here is an example of a promptless install of the PageGate server if the PGSetup folder was copied to C:\

```
C:\PGSetup\pgssetup.exe /silent /dir="c:\PageGate"
```

That command line would install the PageGate server to c:\PageGate\ without prompting for information, would display the progress screens and would prompt to reboot the system if necessary.

Optional Registry Settings

Note: Folders that are set must be created and given sufficient rights before the program is run.

Location: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NotePage\PageGate\

Name	Type	Description
Prog_Dir	String	This value specifies the directory in which the program resides. The path specified in this field must end in a backslash, \.
DB_Dir	String	This value specifies the directory in which the database resides. The path specified in this field must end in a backslash, \.
Log_Dir	String	This value specifies the directory in which the logs reside. The path specified in this field must end in a backslash, \.
Archive_Dir	String	This value specifies the directory in which archived information should reside. The path specified in this field must end in a backslash, \.
DB_Password	String	If a database password has been set on the PageGate database, the password for the database can be specified here. Due to the nature of the security involved, the value is encrypted, so it must be copied from a system with the password already set.

Upgrading to version 10

Make sure you have a v10 registration key. Previous version registration keys will not work with version 10.

- 1) Make a backup of the PageGate database (pagegate.mdb). By default, this file is located in the c:\PageGateData\Database\ directory.

- 2) If you're using PageGate's GetMail Module, please backup the PageGate Mailboxes sub-directory of the PageGate Program directory.
- 3) If you're using PageGate's GetWeb interface, please backup the web templates/webpages.
 - a) On the computer running the PageGate Server, open the PageGate Admin.
 - b) Go to the Interfaces - GetWeb - Settings section.
 - c) Make a note of the Web Pages Directory.
 - d) Browse to the directory indicated in step c.
 - e) In this directory, there will be a sub-directory called Templates.
 - f) To backup your webpage templates, you could copy the Templates folder to a CD, make a copy on a network location, make a copy in a different folder on the current system, the methods of backing up can vary widely.
- 4) Shut down every part of PageGate. Make sure that none of the PageGate processes or applications, even the PageGate Client on the workstations, are running.
- 5) To verify that everything has been shut down, run the Task Manager in Windows by pressing Ctrl+Alt+Del on your keyboard or by right clicking on the task bar and selecting "Task Manager".
- 6) Once the task manager comes up, browse through the Applications or Processes tab. There should be nothing that begins with PG in either list. If you see something that begins with PG that's still in the list, please highlight it and click on End Task.
- 7) Download the latest PageGate installation file from our website (<https://www.notepage.net> - Click the download button under the PageGate section and save the pgsetup.exe file to your desktop).
- 8) Right click on the pgsetup.exe, left click on "Run As Administrator".
- 9) When it asks which installation you wish to run, the only option you want to select is Install PageGate Server.
- 10) Unless you need to change the installation directory, you can really just click next through the entire routine until it finishes.
- 11) Now that we have version 10 installed, right click the shortcut for 'PageGate Admin' and select "Run As Administrator".
- 12) The Admin will prompt you for the version 10 Registration Key. Please enter it, then click Apply.
- 13) Go to the Program - Settings section.

14) Once in the Settings, you should see a section that says "Run PageGate As". If you are using Windows Server 2016 or 2019, select 'PG Service'. For other operating systems, select 'Windows Service'.

15) Click Apply.

When you click on Apply, you will be prompted for a username and password. In most cases, you can leave both values blank and click 'Apply' again.

To explain, if the default System Account in Windows is enabled, you can leave the credentials blank. If the default system account is disabled, you will need to provide an account name and password that has the "Log on as Service" security policy.

[Click here](#) for more information.

16) Open the Services list in Windows. You can do this by either going in to the Services section of the Administrative Tools in the Control Panel or by running services.msc in the Run command.

17) Browse to the PageGate Service.

18) Right click on the PageGate service, tell it to start.

To upgrade the PageGate Client to v10, please refer to [this portion of the manual](#).

To upgrade the Additional PageGate Admin to v10, please refer to [this portion of the manual](#).

To upgrade the Additional PageGate Monitor to v10, please refer to [this portion of the manual](#).

Windows Server 2016 and 2019 Core

These are the steps to install the PageGate Server in Windows Server 2016 and 2019 Core (no GUI).

1) Configure networking: sconfig (this is optional)

2) Run powershell: powershell

3) Download PageGate setup program: invoke-webrequest

`https://www.notepage.net/download/pgsetup.exe -outfile pgsetup.exe`

4) Return to command prompt: exit

5) Run PageGate setup: pgsetup

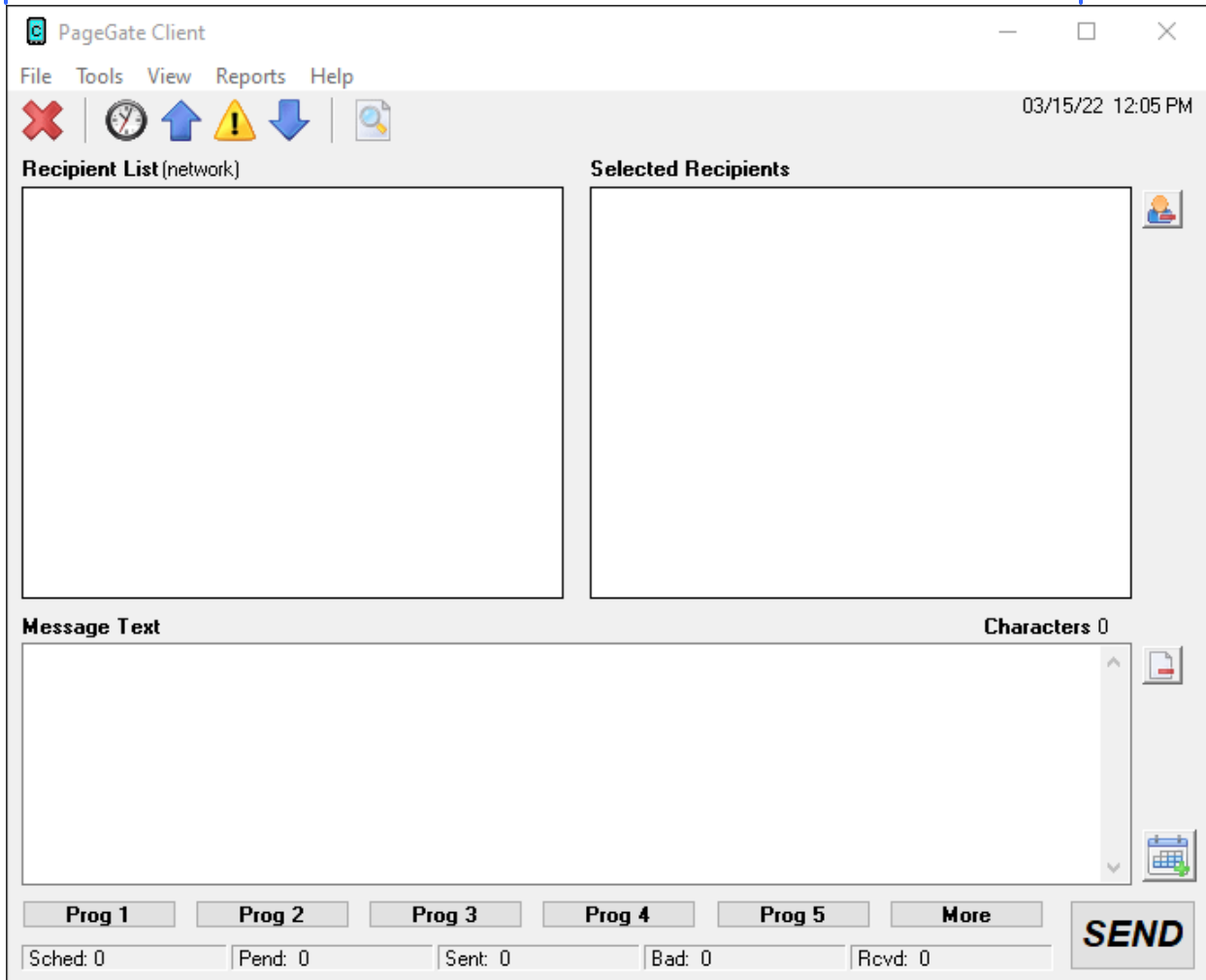
6) select the Install PageGate Server option.

7) Click Next.

- 8) Select whether or not you want to view the ReadMe File and/or Release Notes, then click on Next.
- 9) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click on Next.
- 10) In the next step, you will be prompted for the PageGate Server installation directory. By default, the installation directory is C:\Program Files(x86)\NotePage\PageGate\
The installation directory can be changed to any local path. Click Next to proceed.
- 11) In the next step, changes to the program group PageGate will appear under in the start menu can be made. Click Next to proceed.
- 12) Click Next.
- 13) Click Install to begin installing the PageGate Server.
- 14) Click Finish to finalize the installation.

PageGate GUI Client

The PageGate GUI Client is a Windows based program that can be installed on workstations on the same network as the PageGate server. This program provides a graphical user interface that shows a list of recipients available to message and grants the ability to immediately send or schedule messages.



The PageGate Client can be installed on the server but is intended to be run from workstation computers.

Note: If you have the GUI Client installed on the PageGate server, do not have the workstations reference the PageGate Server's GUI Client program folder. Each workstation requires its own installation of the GUI Client.

Minimum system requirements to run the PageGate GUI Client:

Operating System: Windows 7, Windows 8, Windows 10, Server 2012, Server 2016, Server 2019, Server 2022

Processor: 1.0 Ghz Processor or better.

Memory: 1GB RAM

Storage: 100 MB of free disk space (for application and database)

Please reference the [Installation Steps](#) section of the manual for information on how to install the program.

Pre-installation

There are a few things you should verify before installing the PageGate GUI Client:

- 1) The PageGate Server must be installed and operational before the PageGate GUI Client is installed.
- 2) The PageGate Server's database directory must be shared on the network and the workstation must have sufficient network permissions to the database (pagegate.mdb). The workstation must have read, write, and change permissions to this file.
- 3) If you plan to use the spell-checking feature, Microsoft Office 2000 or newer must be installed on the workstation.
- 4) If application or database passwords have been defined in the PG Admin program, they will be needed to access the application and the PageGate database.

Installation Steps

- [Downloaded installation file](#)

The classic installation method. Download an executable, run it and go through the installation steps.

- [Promptless Automatic installation](#)

A more advanced installation method for deploying the PageGate Client to multiple workstations at once.

- [Upgrading from a previous version](#)

Have a previous version? These steps will guide you through the process to upgrade your existing installation to version 10.

Installation by Download

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>

- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, save the file somewhere convenient to reference, like the Desktop.
- 5) After the file has been downloaded, right click on it and select "Run As Administrator"
- 6) If prompted by the User Accounts Control, allows the installation to proceed.
- 7) When prompted, select Install PageGate GUI Client.
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click on Next.
- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click on Next.
- 11) In the next step, you will be prompted for the PageGate Client installation directory. By default, the installation directory is C:\Program Files (x86)\NotePage\PageGate Client. Once you've specified the installation path, click on Next to proceed.
- 12) In the next step, changes to the program group in the Start Menu the PageGate Client will appear under can be made. Click Next to proceed.
- 13) Select whether you want to create a Desktop Icon for the PageGate Client and click on Next.
- 14) Click Install to begin installing the PageGate Client.
- 15) Click Finish to finalize the installation.

Be sure to make a copy of the installation file and store it somewhere safe. You will need it again if you ever need to reinstall PageGate, or want to install the Additional PageGate Administrator, Additional PageGate Monitor, or GUI Client on a workstation.

Promptless Automatic Installation

Any part of PageGate can be installed unattended. This helps network administrators to automate the program's installation.

- 1) Start the PageGate setup program as usual ([see above](#)).
- 2) When the list of installation options screen is shown, exit the installation.

- 3) Browse in to the c:\Users\\AppData\Local\Temp\ directory.
- 4) Copy the 'PGSetup' folder to another location.
- 5) Use the following command line syntax to run a promptless install.

Installation executable for the GUI Client: pgcsetup.exe

Command line options:

/silent	Installs without prompts. Does display progress screens, will prompt for reboot if necessary.
/verysilent	Installs without prompts, display screens and will reboot without prompting, if necessary.
/norestart	This flag prevents the installation from rebooting, even if necessary.
/dir ="x:\dirname"	This flag overrides the default installation directory.
/group="folder name"	This flag overrides the default Windows Start Menu group name.
/noicons	This flag will prevent the installation from creating the desktop icons for the program.

Here is an example of a promptless install of the PageGate server if the PGSetup folder was copied to C:\

```
C:\PGSetup\pgcsetup.exe /silent /dir="c:\PageGate Client"
```

That command line would install the PageGate server to c:\PageGate Client\ without prompting for information, would display the progress screens and would prompt to reboot the system if necessary.

Optional Registry Settings

Client Registry Settings

Note: Folders that are set must be created and given sufficient rights before the program is run.

Location: HKEY_CURRENT_USER\SOFTWARE\NotePage\NotePagerNet\

Name	Type	Description
LockSettings	Dword	By default settings in the Current User portion of the registry are used. If every user on the workstation should use the standardized, rather than logon specific, settings, then the Local Machine portion of the registry can be forced to be used instead by setting LockSettings to 1, which is a 'true' value.

DB_Dir	String	This value specifies the directory in which the database resides. The path specified in this field must end in a backslash, \.
DB_Password	String	If a database password has been set on the PageGate database, the password for the database can be specified here. Due to the nature of the security involved, the value is encrypted, so it must be copied from a system with the password already set.
Sender	String	This value specifies a static sender's name that this copy of the Client will use when sending messages.
Station	String	Note: Values lower than 10 are system reserved and should never be used. Clients with the same Station value will be able to reference each others' message histories and be able to run reports on the same.

Upgrading from a previous version

Make sure you **DO NOT** have the PageGate Client running. The update cannot be applied if the PG Client is running.

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>
- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, you want to save the file.
- 5) Save the pgsetup.exe file somewhere easy to reference, like the Desktop.
- 6) After the file has been downloaded, right click on it and select "Run As Administrator".
- 7) When prompted, select: Install PageGate GUI Client
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click Next.
- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click Next.
- 11) Click Install to begin installing the version 10 PageGate GUI Client.
- 12) Click Finish to finalize the installation.

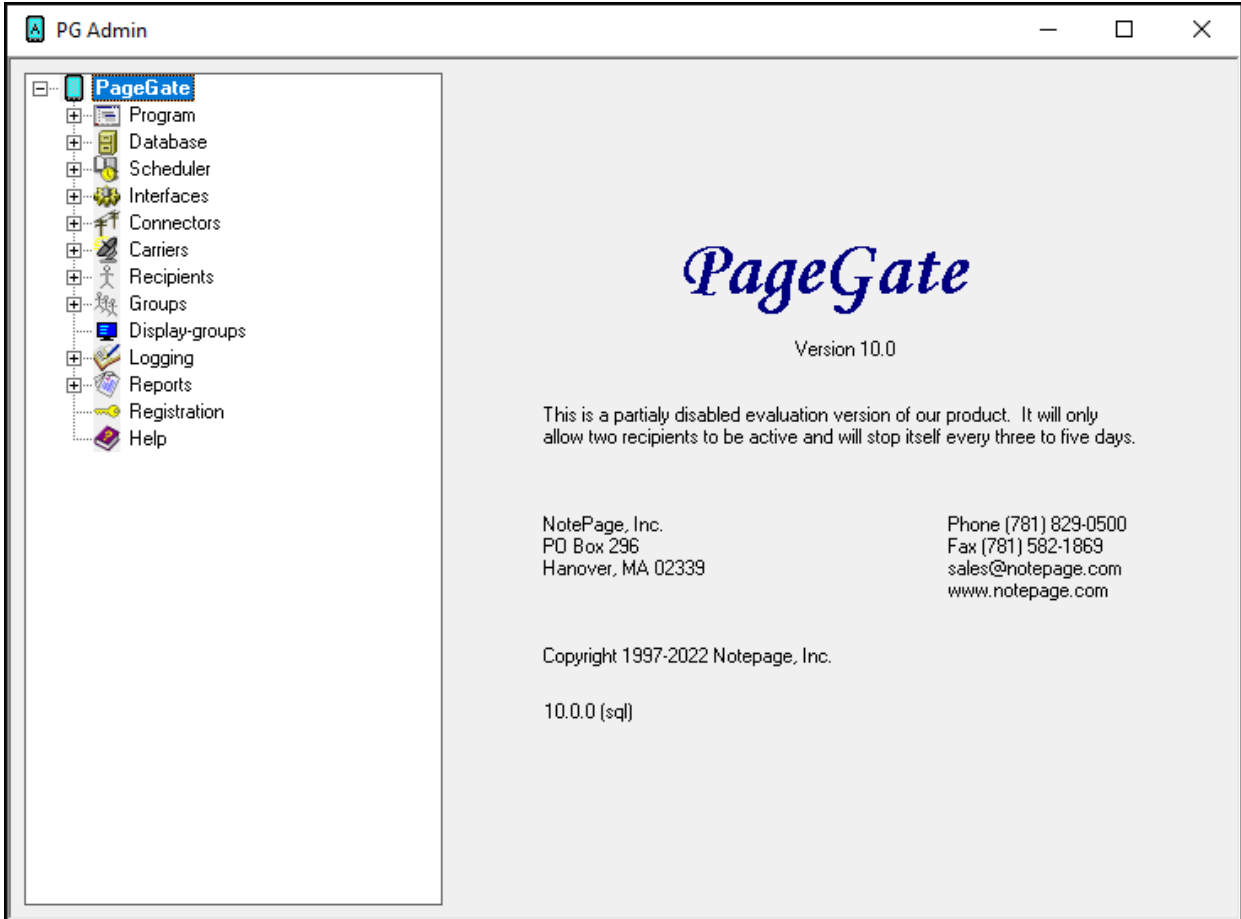
Installation

4

Additional PG Admin

Do not install this on the PageGate Server. The Additional PageGate Admin is only intended for workstations.

The Additional PageGate Admin is a program that allows a workstation the capability to modify PageGate's settings, such as the ability to add, modify, or remove recipients and carriers. The workstation intended to run the Additional PageGate Admin must have network access to the PageGate Server's database.



Minimum system requirements to run the PageGate Admin

Operating System: Windows 7, Windows 8, Windows 10, Server 2012, Server 2016, Server 2019, Server 2022

Processor: 1.0 Ghz Processor or better.

Memory: 1GB RAM

Storage: 100 MB of free disk space (for application and database)

Please reference the [Installation Steps](#) section of the manual for information on how to install the program.

Pre-installation

There are a few things that should be verified and setup before the installation of PG

Admin:

- 1) Do not install this on the PageGate Server. The Additional PageGate Admin program is intended only for workstations.
- 2) Be sure your hardware meets the minimum specs for PageGate Admin.
- 3) Be sure the computer is configured for your network. The PageGate Admin must access the PageGate database through your LAN (Local Area Network).
- 4) Be sure that you have sufficient permissions (if applicable) to install new programs on the chosen computer.
- 5) In some cases, you will be asked to restart your computer after the installation of the PageGate Admin. If so, this must be done before the PageGate Admin is run. Be sure that you install the PageGate Admin at a time when this can be done.
- 6) Always be sure to have a complete backup of your system before installing any new software! The installation of the PageGate Admin is not an exception to this rule.

Installation Steps

- [Downloaded installation file](#)

The classic installation method. Download an executable, run it and go through the installation steps.

- [Promptless Automatic installation](#)

A more advanced installation method for deploying the PageGate Admin to multiple workstations at once.

- [Upgrading to version 10 from a previous version](#)

Have a previous version (4 - 9)? These steps will guide you through the process to upgrade your existing installation to version 10.

Installation by Download

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>
- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, you want to save the file.
- 5) Save the pgsetup.exe file somewhere easy to reference, like the Desktop.
- 6) After the file has been downloaded, right click on it and select "Run As Administrator".
- 7) When prompted, select: Install Additional PG Admin
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click on Next.
- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click on Next.
- 11) In the next step, you will be prompted for the PageGate Admin installation directory. By default, the installation directory is C:\Program Files(x86)\NotePage\PageGate Admin\ The installation directory can be changed to any local path. Click Next to proceed.
- 12) In the next step, changes to the program group for the PageGate Admin will appear under in the start menu can be made. Click Next to proceed.
- 13) Select whether you want to create a Desktop Icon for the PageGate Admin and click on Next.
- 14) Click Install to begin installing the PageGate Admin.
- 15) Click Finish to finalize the installation.

Promptless Automatic Installation

Any part of PageGate can be installed unattended. This helps network administrators to automate the program's installation.

- 1) Start the PageGate setup program as usual ([see above](#)).
- 2) When the list of installation options screen is shown, exit the installation.

- 3) Browse in to the c:\Users\\AppData\Local\Temp\ directory.
- 4) Copy the 'PGSetup' folder to another location.
- 5) Use the following command line syntax to run a promptless install.

PageGate Server Installation File: pgasetup.exe

Command line options:

/silent	Installs without prompts. Does display progress screens, will prompt for reboot if necessary.
/verysilent	Installs without prompts, display screens and will reboot without prompting, if necessary.
/norestart	This flag prevents the installation from rebooting, even if necessary.
/dir="x:\dirname"	This flag overrides the default installation directory.
/group="folder name"	This flag overrides the default Windows Start Menu group name.
/noicons	This flag will prevent the installation from creating the desktop icons for the program.

The <path> option should be the 'PGSetup' folder in step #4 above.

Here is an example of a promptless install of the PageGate Admin if the PGSetup folder was copied to C:\

```
C:\PGSetup\pgasetup.exe /silent /dir="c:\PageGate Admin"
```

That command line would install the PageGate server to c:\PageGate Admin\ without prompting for information, would display the progress screens and would prompt to reboot the system if necessary.

Optional Registry Settings

Note: Folders that are set must be created and given sufficient rights before the program is run.

Location: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NotePage\PageGate\

Name	Type	Description
Prog_Dir	String	This value specifies the directory in which the program resides. The path specified in this field must end in a backslash, \.
DB_Dir	String	This value specifies the directory in which the database resides. The path specified in this field must end in a backslash, \.

Log_Dir	String	This value specifies the directory in which the logs reside. The path specified in this field must end in a backslash, \.
Archive_Dir	String	This value specifies the directory in which archived information should reside. The path specified in this field must end in a backslash, \.
DB_Password	String	If a database password has been set on the PageGate database, the password for the database can be specified here. Due to the nature of the security involved, the value is encrypted, so it must be copied from a system with the password already set.

Upgrading from a previous version

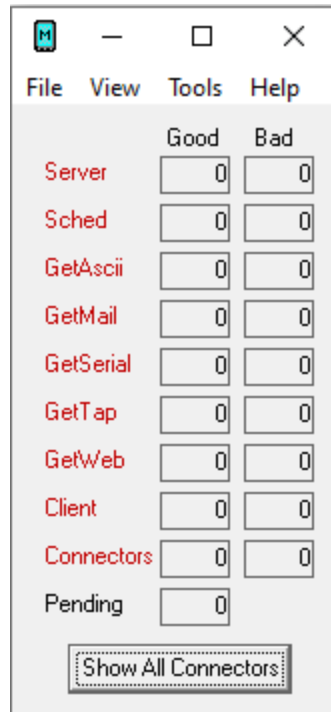
Make sure you DO NOT have the PageGate Admin running. The update cannot be applied if the PG Admin is running.

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>
- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, you want to save the file.
- 5) Save the pgsetup.exe file somewhere easy to reference, like the Desktop.
- 6) After the file has been downloaded, right click on it and select "Run As Administrator".
- 7) When prompted, select: Install Additional PG Admin
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click Next.
- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click Next.
- 11) Click Install to begin installing the version 10 PageGate PG Admin.
- 12) Click Finish to finalize the installation.

Additional PG Monitor

Do not install this on the PageGate Server. The Additional PageGate Monitor program is only intended for workstations.

The Additional PageGate Monitor is a program that allows a workstation to view what's happening in PageGate as it happens. It shows which modules are running, how many messages have been processed, how many of the processed messages are good, how many of the processed messages are bad, and also gives the detailed log of the precise transactions between PageGate, its modules and the messaging hosts PageGate contacts.



Minimum requirements to run the PageGate Monitor

Operating System: Windows 7, Windows 8, Windows 10, Server 2012,

Server 2016, Server 2019, Server 2022

Processor: 1.0 Ghz Processor or better.

Memory: 1GB RAM

Storage: 100 MB of free disk space (for application and database)

Please reference the [Windows 8, 10, Server 2008, Server 2012, Server 2016 and Server 2019](#) portion of the manual to install on those operating systems.

Installation Steps

- [Installation by download](#)

The classic installation method. Download an executable, run it and go through the installation steps.

- [Promptless Automatic installation](#)

A more advanced installation method for deploying the PageGate Monitor to multiple workstations at once.

- [Upgrading from a previous version](#)

Have a previous version? These steps will guide you through the process to upgrade your existing installation to version 10.

Installation by Download

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>
- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, you want to save the file.
- 5) Save the pgsetup.exe file somewhere easy to reference, like the Desktop.
- 6) After the file has been downloaded, right click on it and select "Run As Administrator".
- 7) When prompted, select: Install Additional PG Monitor
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click on Next.
- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click on Next.
- 11) In the next step, you will be prompted for the installation directory. By default, the installation directory is C:\Program Files(x86)\NotePage\PageGate Monitor\
The installation directory can be changed to any local path. Click Next to proceed.
- 12) In the next step, changes to the program group PageGate will appear under in the start menu can be made. Click Next to proceed.

13) Select whether you want to create a Desktop Icon for the PageGate Monitor and click on Next.

14) Click Install to begin installing the PageGate Monitor.

15) Click Finish to finalize the installation.

Promptless Automatic Installation

Any part of PageGate can be installed unattended. This helps network administrators to automate the program's installation.

- 1) Start the PageGate setup program as usual ([see above](#)).
- 2) When the list of installation options screen is shown, exit the installation.
- 3) Browse in to the c:\Users\\AppData\Local\Temp\ directory.
- 4) Copy the 'PGSetup' folder to another location.
- 5) Use the following command line syntax to run a promptless install.

PageGate Server Installation File: pgmsetup.exe

Command line options:

/silent	Installs without prompts. Does display progress screens, will prompt for reboot if necessary.
/verysilent	Installs without prompts, display screens and will reboot without prompting, if necessary.
/norestart	This flag prevents the installation from rebooting, even if necessary.
/dir ="x:\dirname"	This flag overrides the default installation directory.
/group="folder name"	This flag overrides the default Windows Start Menu group name.
/noicons	This flag will prevent the installation from creating the desktop icons for the program.

The <path> option should be the 'PGSetup' folder in step #4 above.

Here is an example of a promptless install of the PageGate Monitor if the PGSetup folder was copied to C:\

```
C:\PGSetup\pgmsetup.exe /silent /dir="c:\PageGate Monitor"
```

That command line would install the PageGate server to c:\PageGate Monitor\ without prompting for information, would display the progress screens and would prompt to reboot the system if necessary.

Optional Registry Settings

Location: HKEY_LOCAL_MACHINE\SOFTWARE\NotePage\PageGate\

Name	Type	Description
PGM_Port	Dword, Decimal	This key specifies the port number that PageGate uses to send updates via UDP. This setting MUST match the setting used by the PageGate Server, referenced in the Logging section of the documentation. Use a value of 10101 decimal, unless the server value has been changed.
NPP_Sender	String	If the PageGate Monitor's NotePager Pro Alerts have been enabled, this key specifies the Sender information for these alerts.

Upgrading from a previous version

Make sure you do not have the PageGate Monitor running. The update cannot be applied if the PG Monitor is running.

- 1) Open a web browser.
- 2) Go to our website: <https://www.notepage.net>
- 3) In the center of the page, click on the Download button under the PageGate section.
- 4) When prompted, you want to save the file.
- 5) Save the pgsetup.exe file somewhere easy to reference, like the Desktop.
- 6) After the file has been downloaded, right click on it and select "Run As Administrator".
- 7) When prompted, select: Install Additional PG Monitor
- 8) Click Next.
- 9) Select whether or not you want to view the ReadMe File and/or Release Notes, then click Next.
- 10) Please read the End User License Agreement (EULA), then select "I accept the agreement" and click Next.
- 11) In the next step, you will be prompted for the installation directory. Specify the existing PageGate Monitor program directory, then click Next to proceed.

- 12) In the next step, changes to the program group PageGate will appear under in the start menu can be made. Click Next to proceed.
- 13) Select whether you want to create a Desktop Icon for the PageGate Monitor and click Next.
- 14) Click Install to begin installing the PageGate Monitor.
- 15) Click Finish to finalize the installation.

To run PageGate for the first time, you'll need to choose your database style, configure PageGate's input interfaces, configure delivery methods, then configure recipients and groups.

Logically speaking, PageGate breaks down in to two halves: input and output. So, you'll need to tell the program how you'll be passing messages in and then how to deliver those messages and who they're going to.

Step 1: Choose your database style

PageGate currently supports two different database styles: SQL and MSJet. PageGate's default installation uses the MSJet database format but you can quite easily configure the program to use SQL instead.

- [Click here for steps to use an MSJet database](#) (default installation)
- [Click here for steps to use an SQL database](#)

Step 2: Configure PageGate to run as a System Service

PageGate can be configured to run in three different styles: Application, Windows Service and PG Service. We **strongly** recommend using the Windows Service or PG Service run styles.

- [Click here for steps to change PageGate's run style to a Windows Service or PG Service.](#)

Step 3: Configure PageGate's input interface(s)

After getting the program installed and the basic database created, you'll need to tell PageGate how you're going to be passing messages into the system.

For more information on that, have a look at the "[How PageGate Accepts Messages](#)" or the "[Interfaces](#)" section of the documentation.

Step 4: Configure your delivery methods

Running PageGate for the First Time

5

Once you have the input interfaces configured, you're ready to start passing messages in to the program for delivery. So, the next step is to tell the program how you're going to be delivering messages.

Message delivery is largely controlled by the type of messages you want to send. Are you sending to cell phones? Email addresses? Making API calls in to a ticket tracking or chat system?

For more detailed information, have a look at the "[How PageGate Sends Messages](#)" or "[Carriers](#)" section of the documentation.

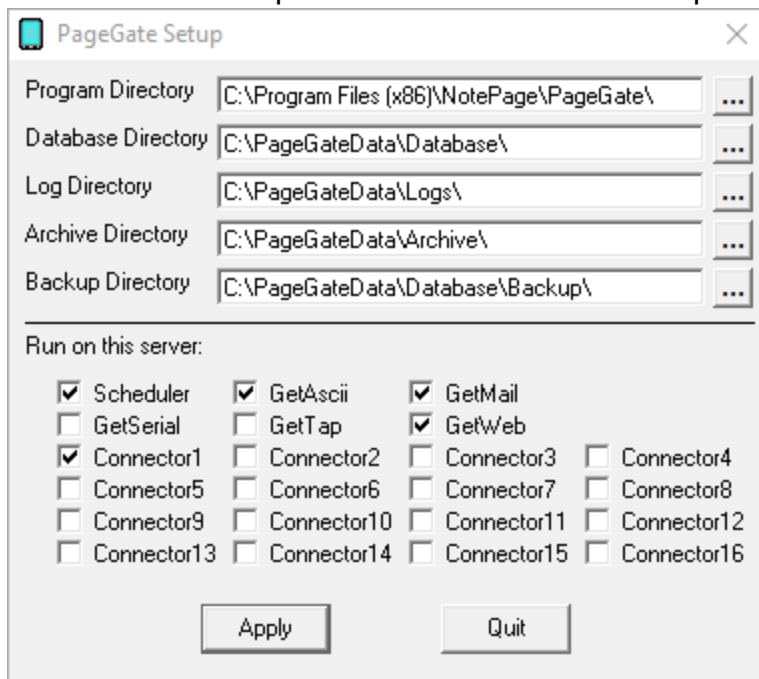
Step 5: Configure your recipients and groups

Now that PageGate knows how messages are going to come in and how to deliver them, you'll need to configure recipients and groups to tell it who the messages are going to.

For more detailed information, have a look at the "[Recipients](#)" and "[Groups](#)" section of the documentation.

Standard Setup (MSJet)

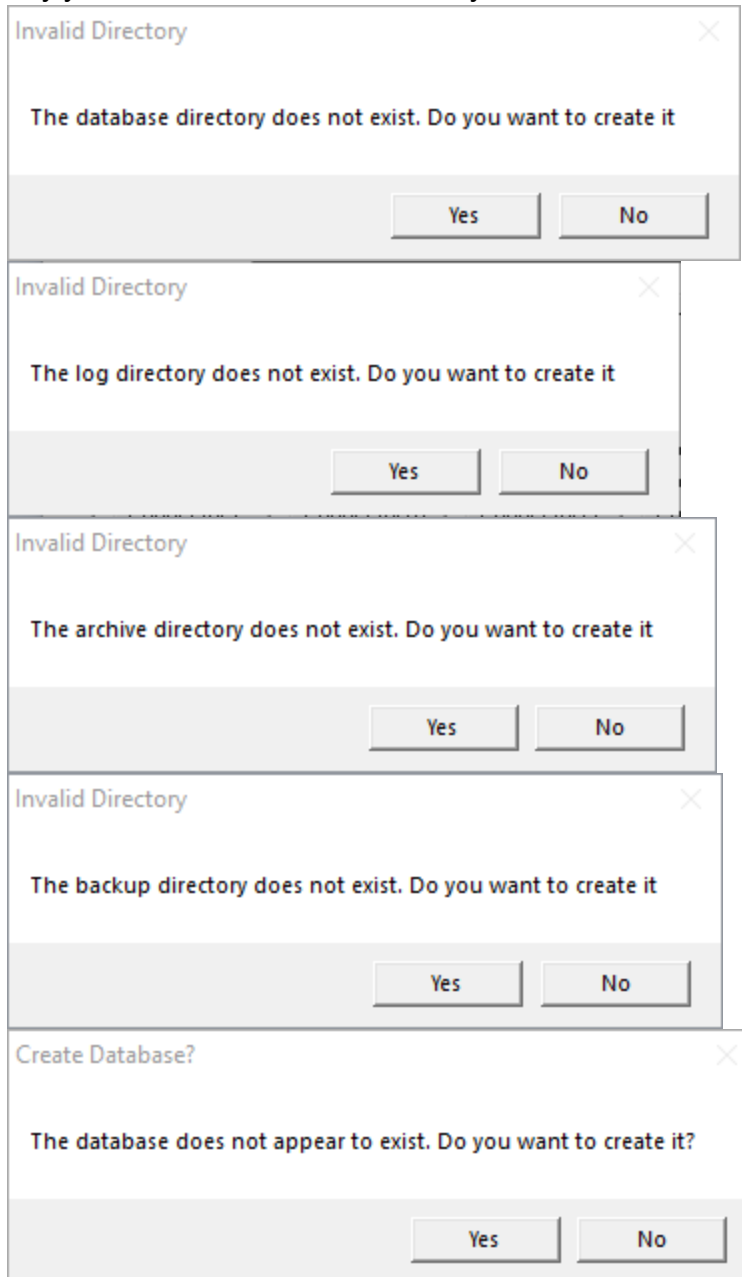
- 1) Go through the [steps to install the PageGate Server](#).
- 2) From your Start Menu or Desktop, right click on the shortcut for the PageGate Admin and select "Run as Administrator"
- 3) You should then be presented with the default setup screen.



You can leave the values at their defaults or specify custom locations. Click Apply to

proceed.

- 4) If it asks you to create the Database, Archive, Log and Backup directories or database, say yes and it will create them for you.



You can now begin configuring the [interfaces](#), [output methods](#) and [recipients](#) and [groups](#).

We also recommend [switching PageGate's run style](#) to PG Service or Windows service.
SQL Setup

PageGate currently supports using SQL Express and SQL Server 2017 and 2019 deployed both on local networks and on cloud based servers.

Running PageGate for the First Time

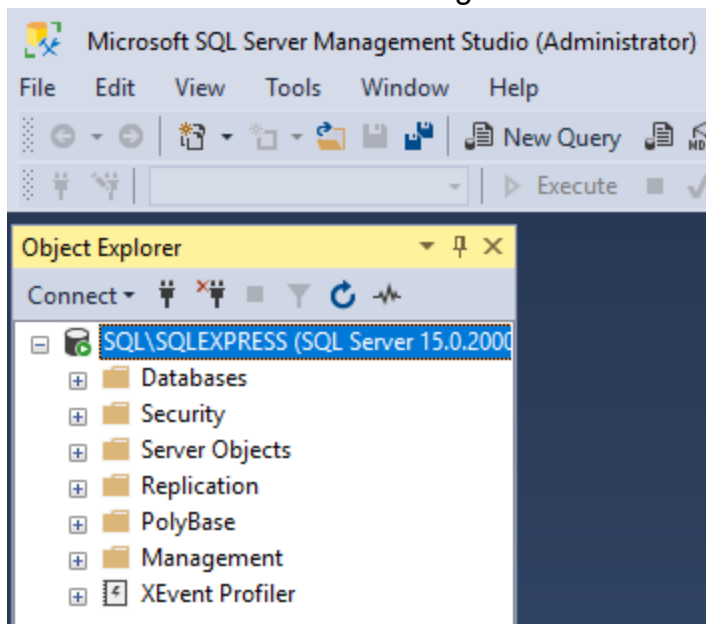
5

When using an SQL database hosted on a cloud server, if you enable 'Always Encrypted' in SQL, the PageGate Admin and GUI Client applications can be installed on any computer and given access to your PageGate server. This makes connecting remote workstations far easier as they can be given direct access.

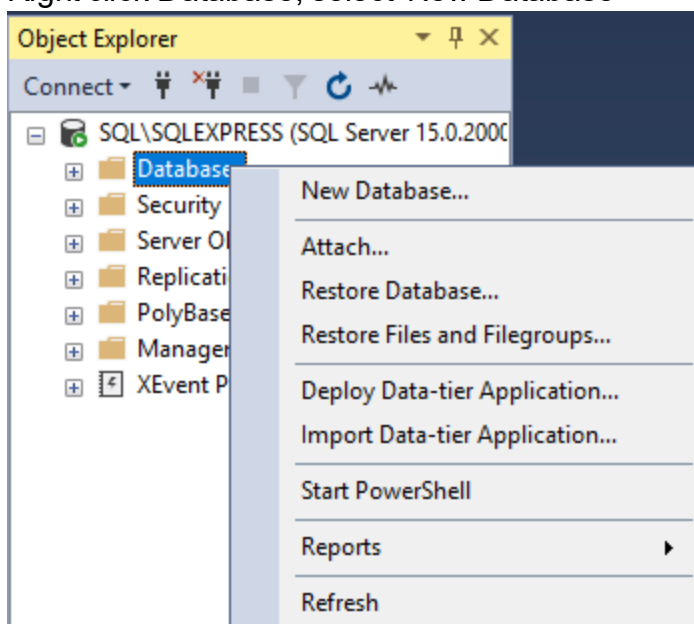
These steps go over how to configure PageGate to use an SQL database.

Step 1: Create the SQL Database

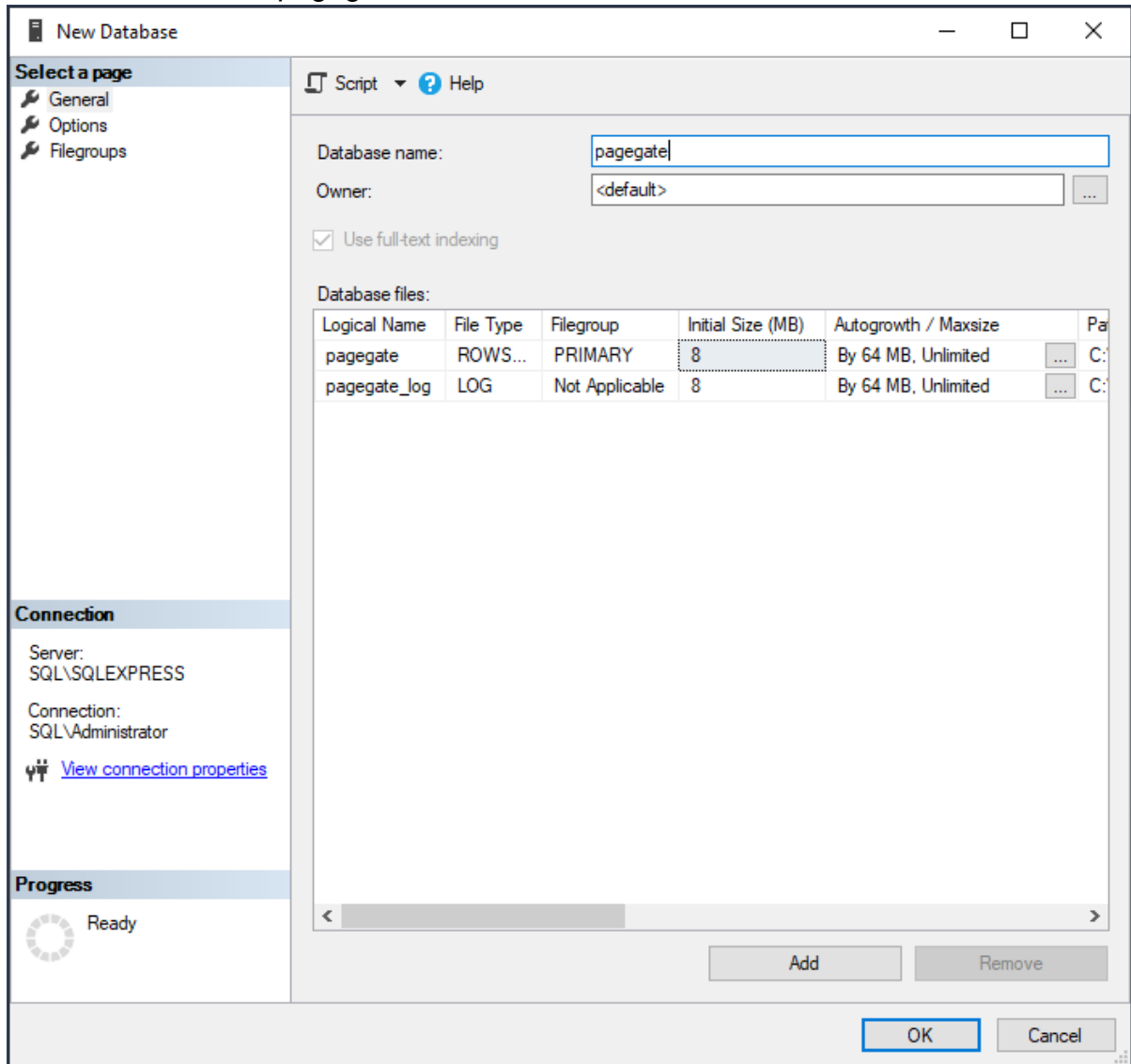
- 1) Run Microsoft SQL Server Management Studio



- 2) Right click Database, select 'New Database'



3) Name the database 'pagegate' and click OK.



Step 2: Create the Admin and User accounts for the SQL Server

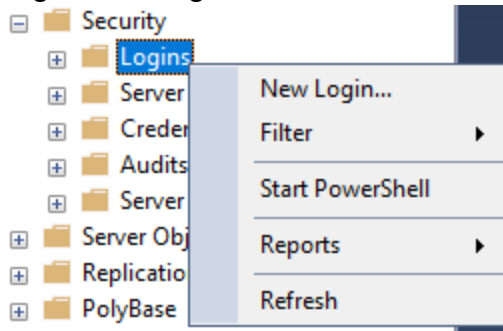
Note: You can skip this step if all relevant users already exist as SQL Logins.

1) Expand the 'Security' section.

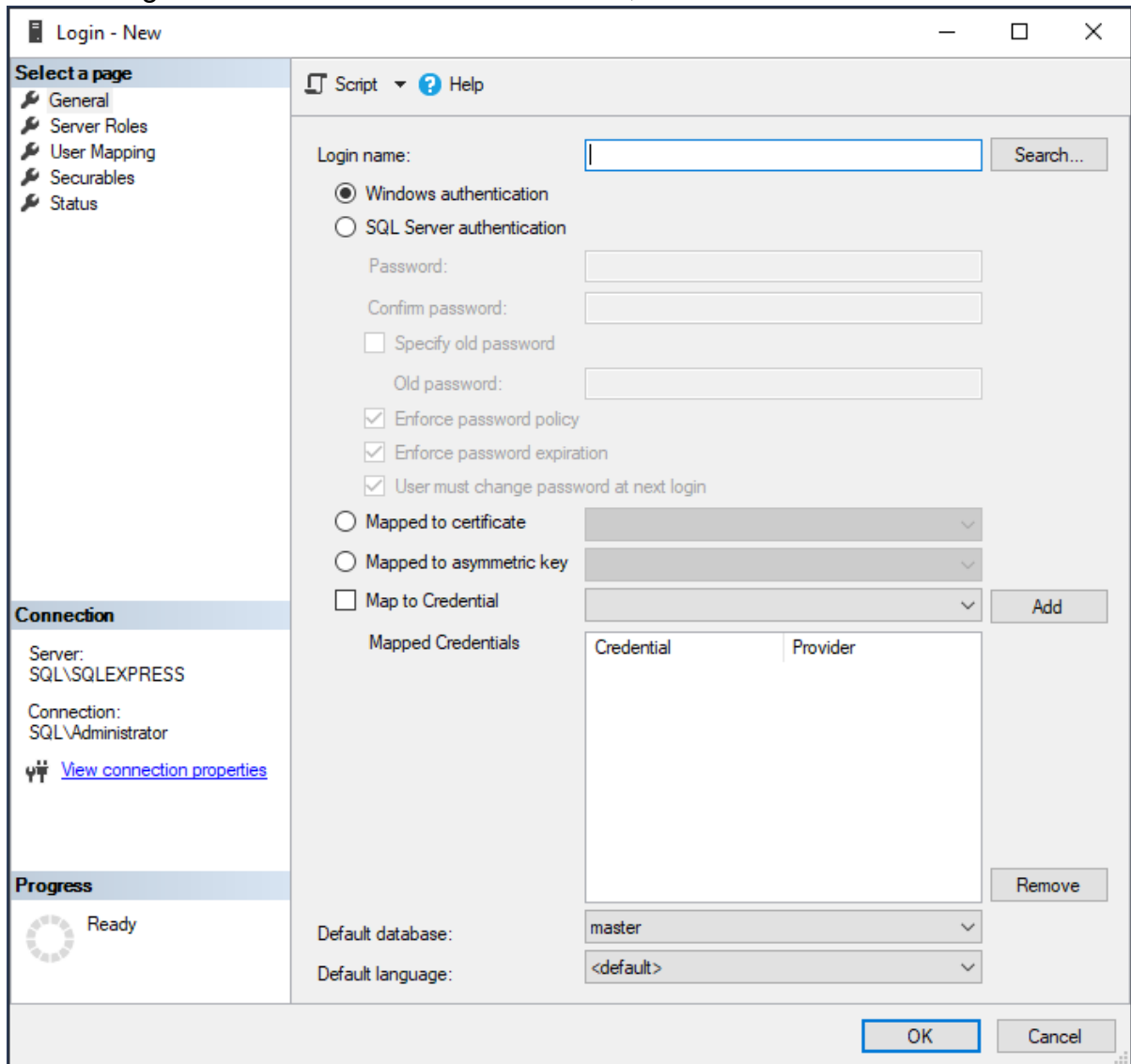
Running PageGate for the First Time

5

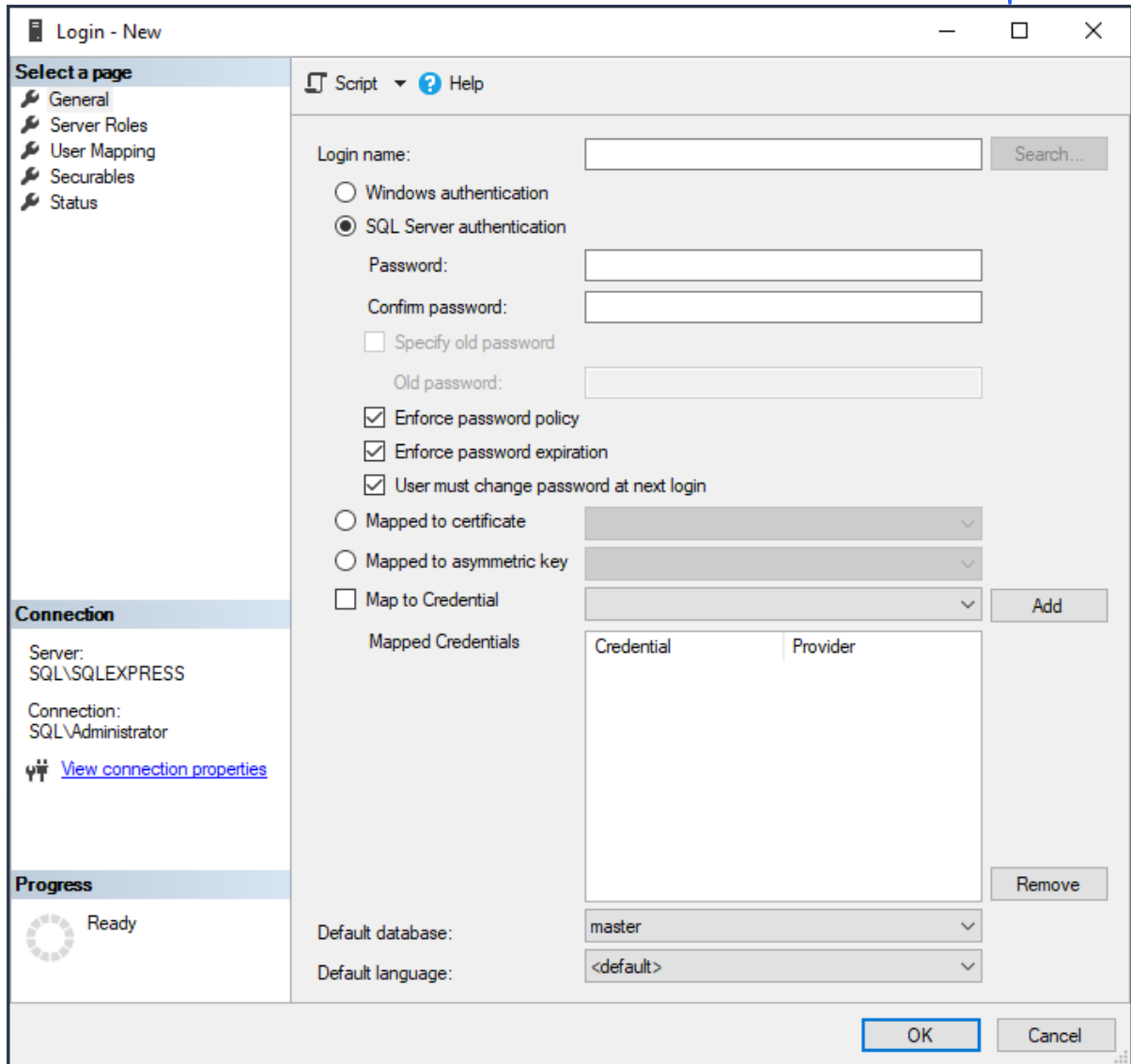
2) Right click Logins and select 'New Login'.



3) When using a domain controller/domain network, select 'Windows authentication'



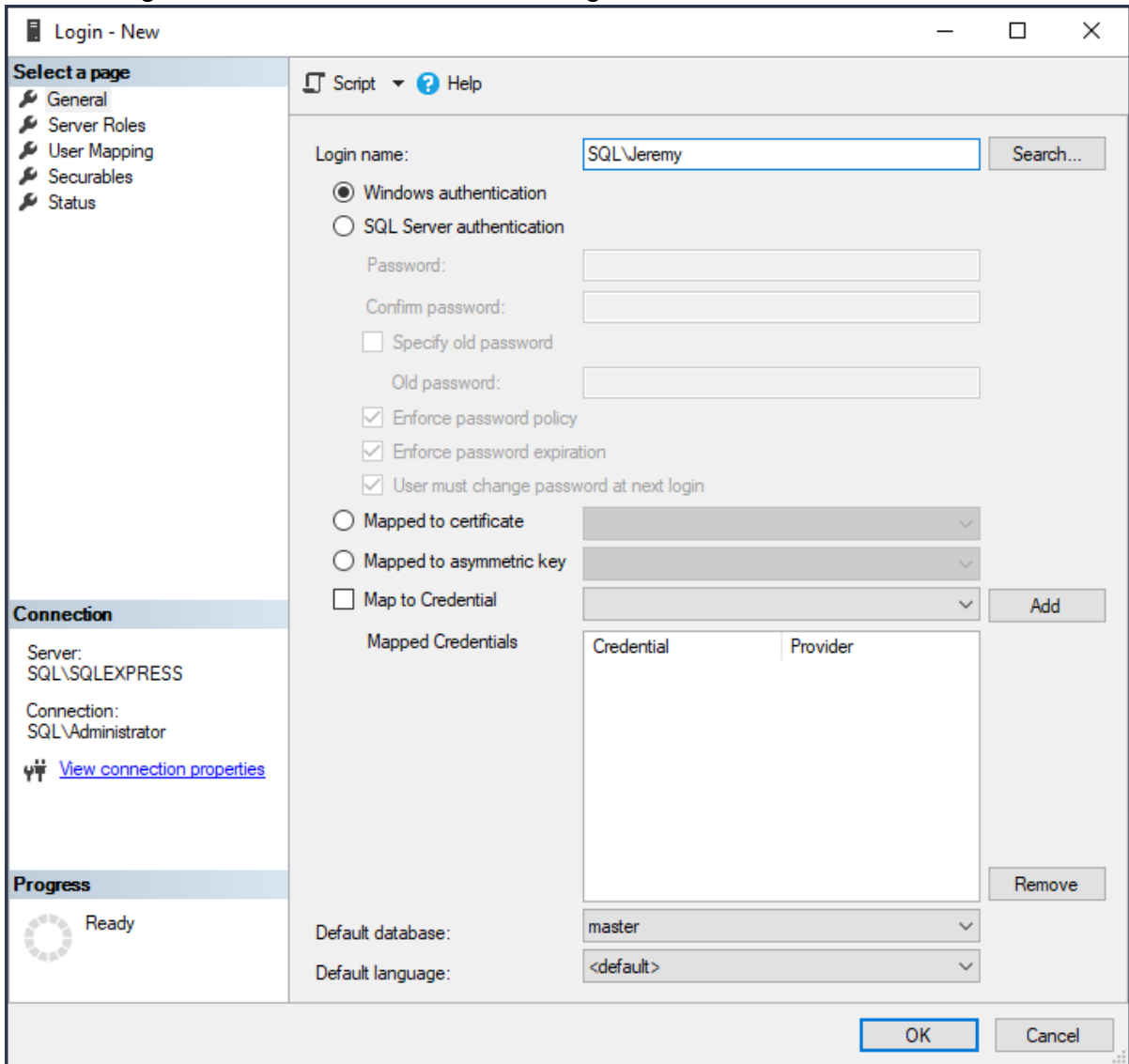
When using a workgroup network, select 'SQL Server authentication'.



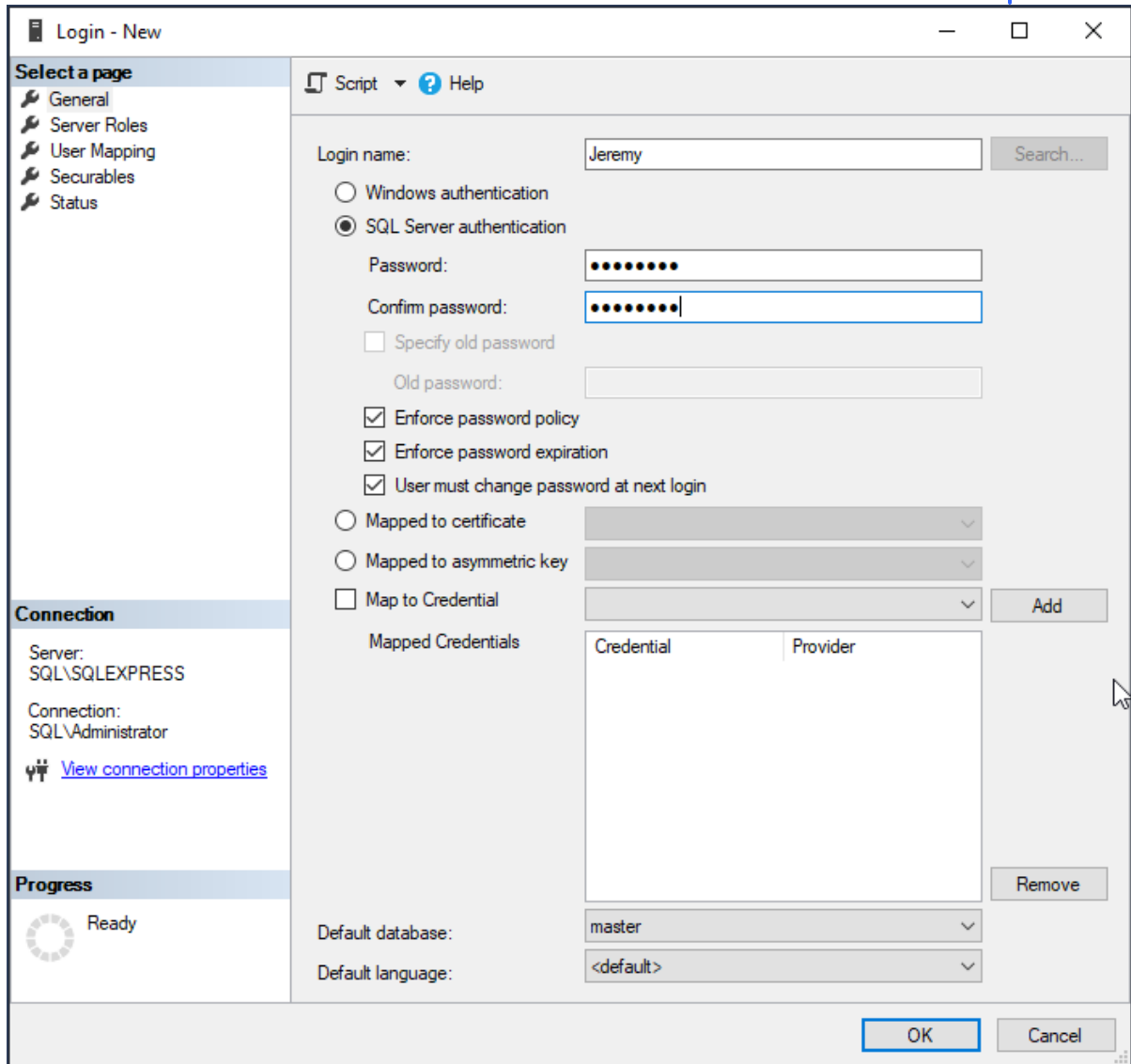
Running PageGate for the First Time

5

4) When using 'Windows authentication', set 'Login name' to a domain account.



When using 'SQL Server authentication', specify the username and password you would like to use.



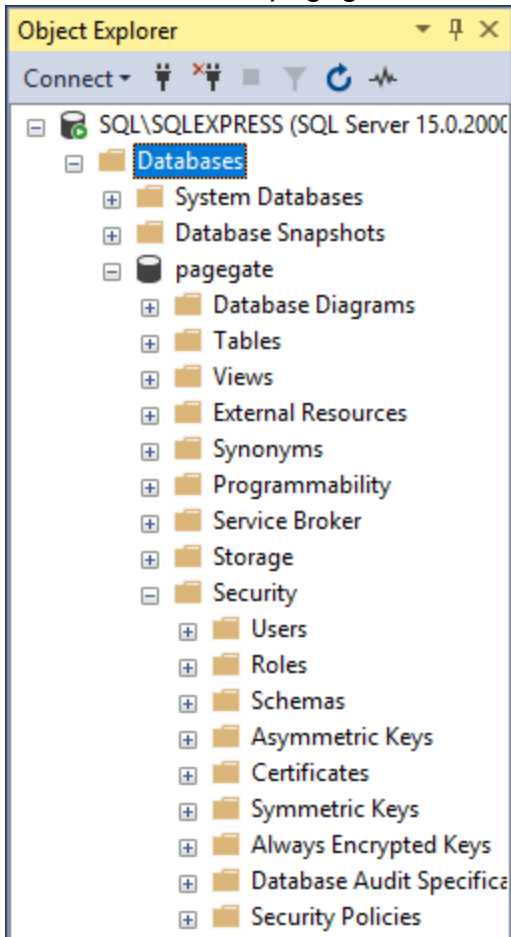
5) Click OK to add the new login.

Step 3: Create the SQL Admin and User accounts for the PageGate database

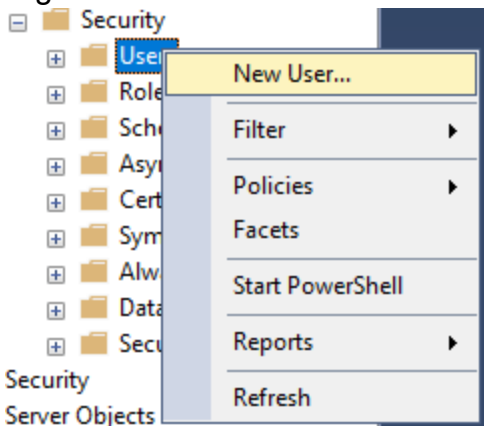
Running PageGate for the First Time

5

1) Go to Databases - pagegate - Security - Users.



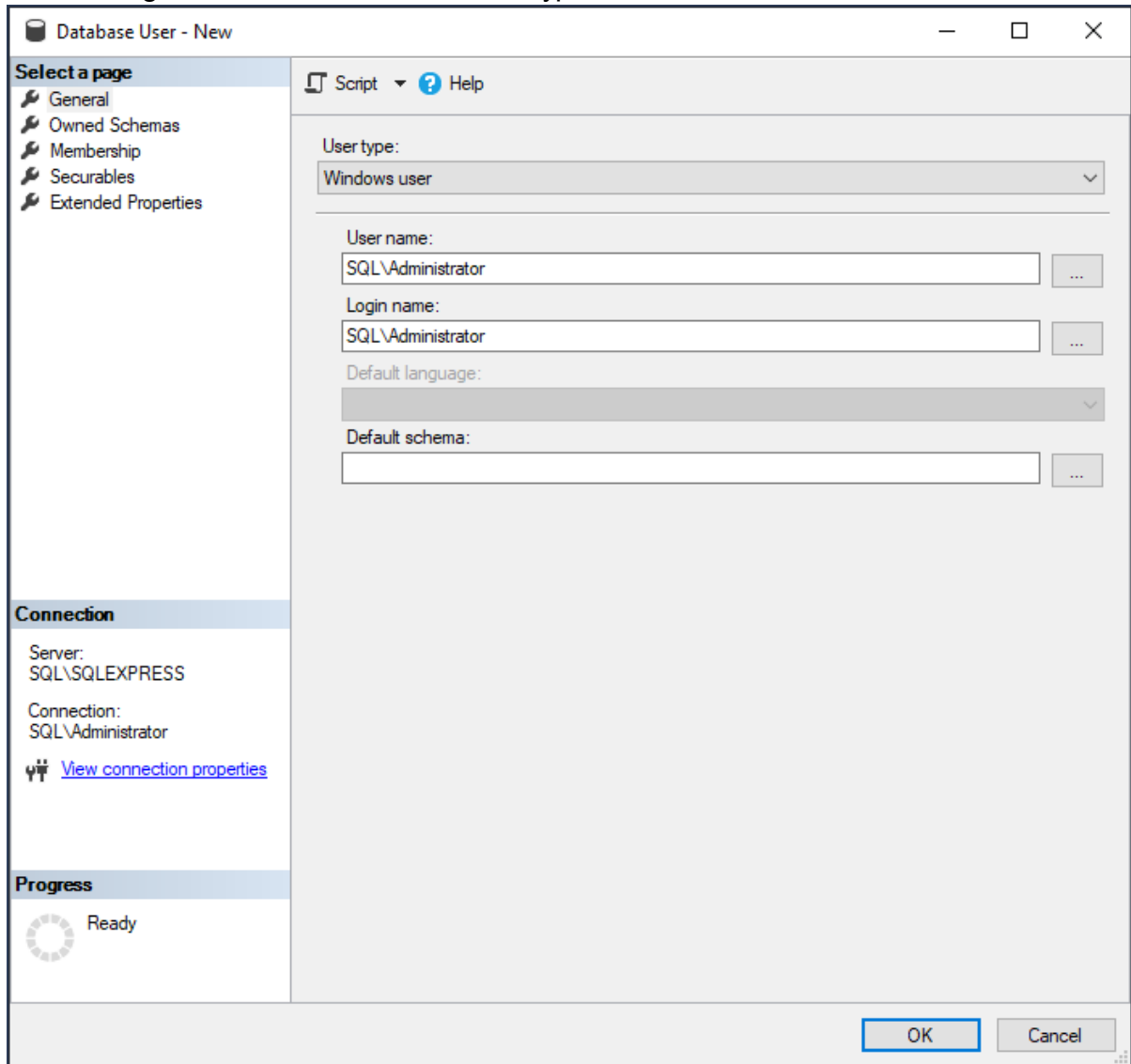
2) Right click Users and select 'New User'.



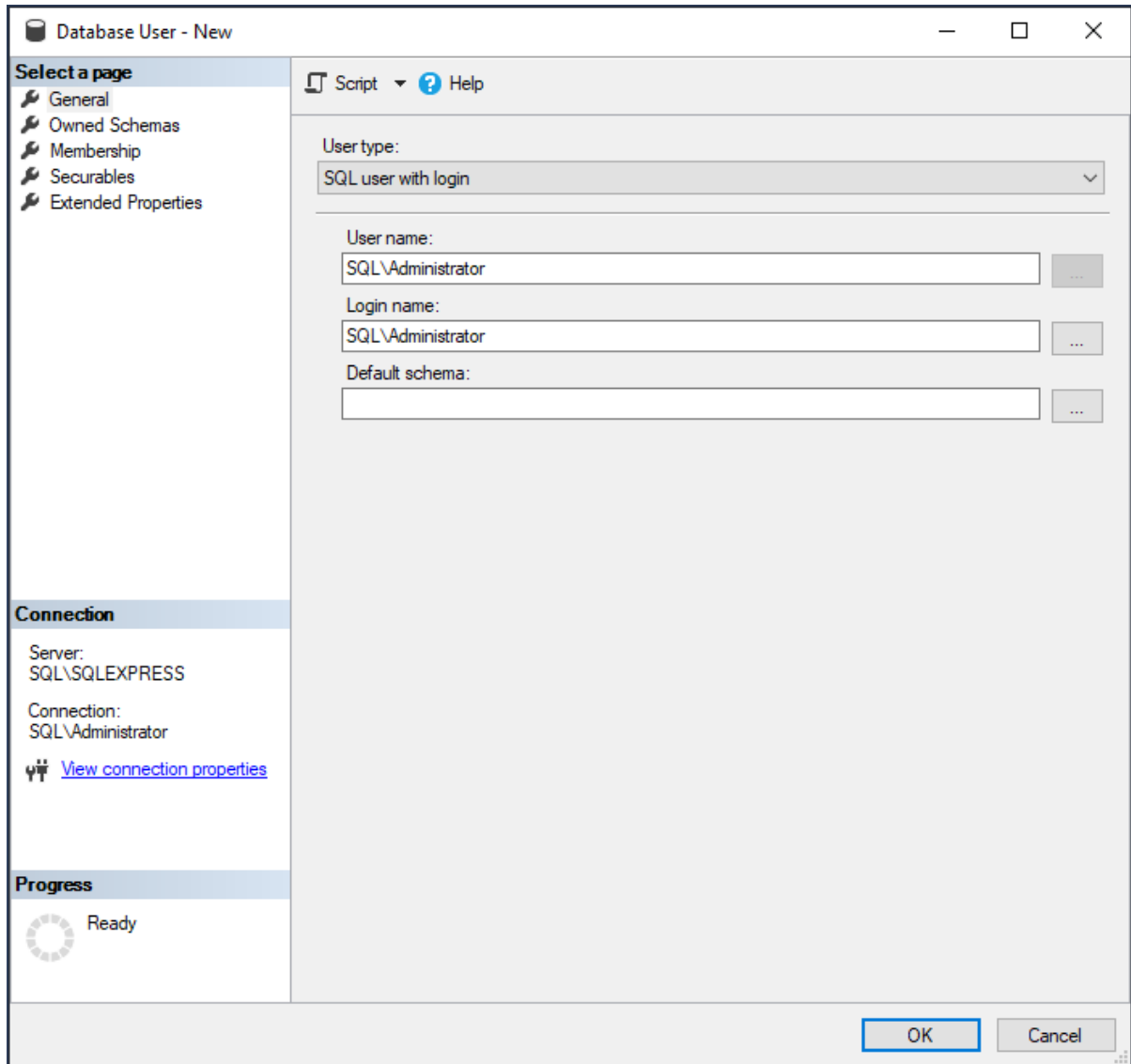
Running PageGate for the First Time

5

6) When using a domain network, set 'User type' to 'Windows authentication'.

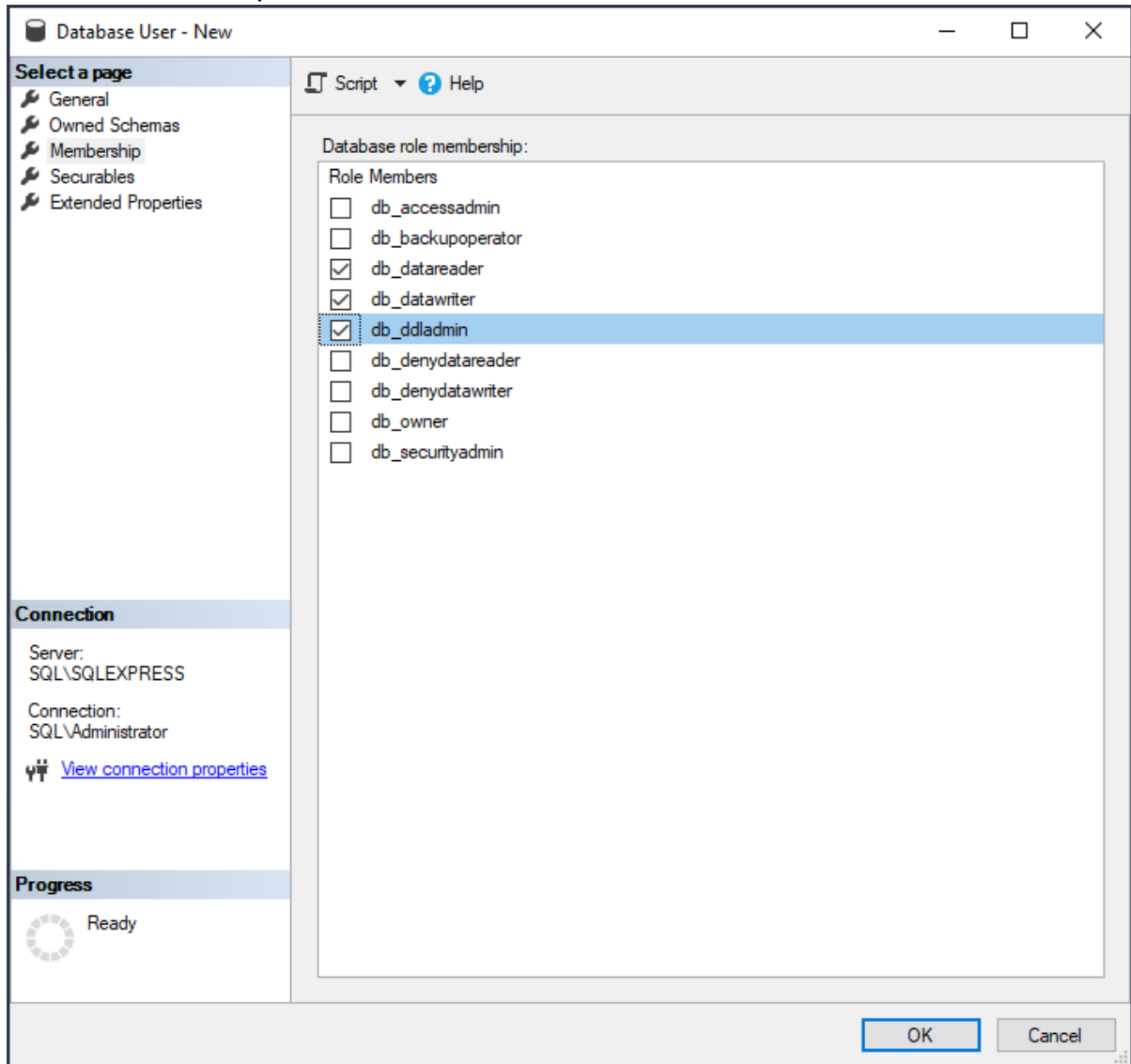


When using a workgroup network, set 'User type' to 'SQL user with login'.



7) Set the username and login name to the account required.

8) Select 'Membership' on the left hand side.



9) Your administrative accounts will need the db_datareader, db_datawriter and db_ddladmin memberships. Your non-administrative accounts will need the db_datareader and db_datawriter memberships.

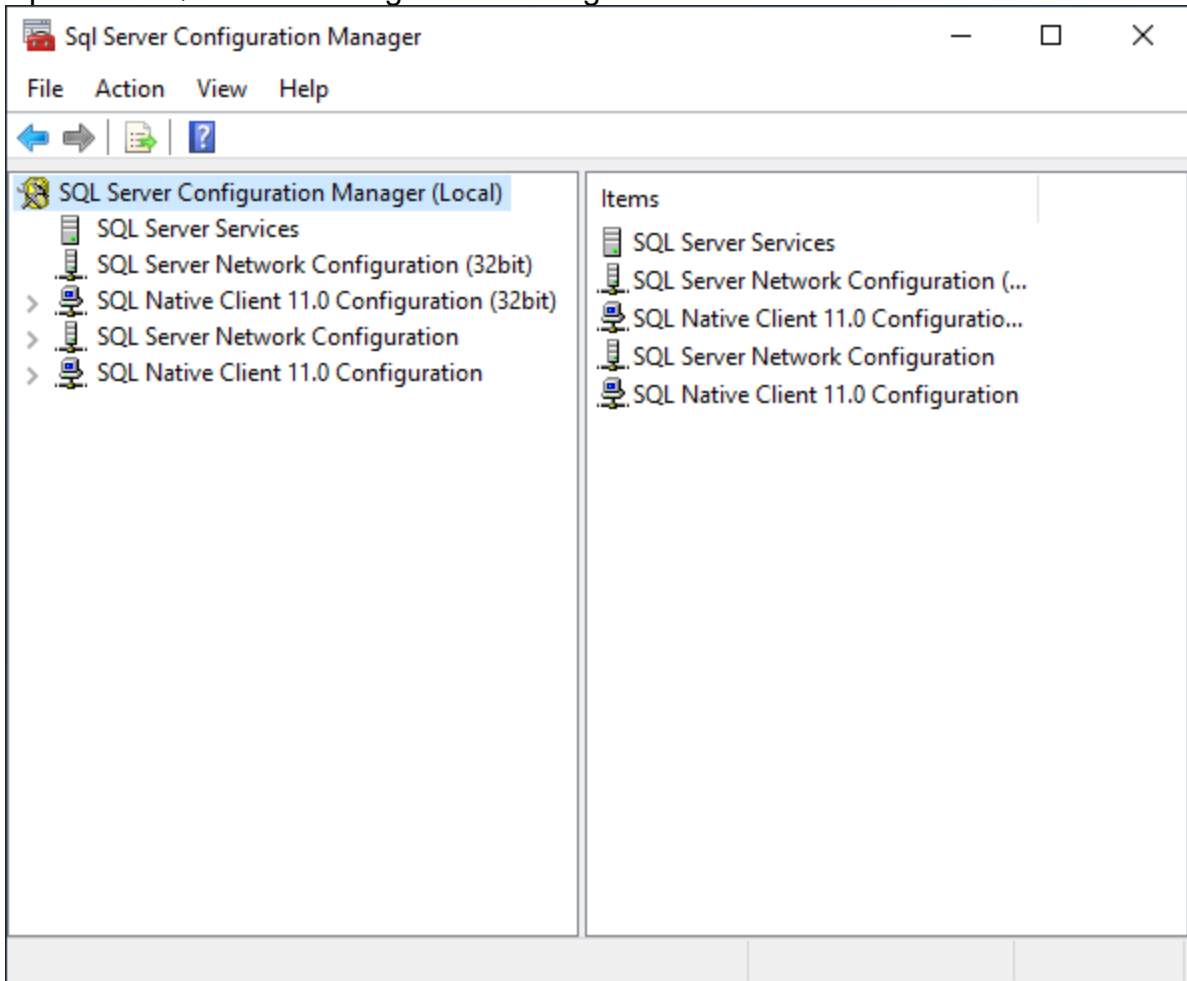
10) Click OK.

Step 4: Configure SQL Server to use encrypted connection

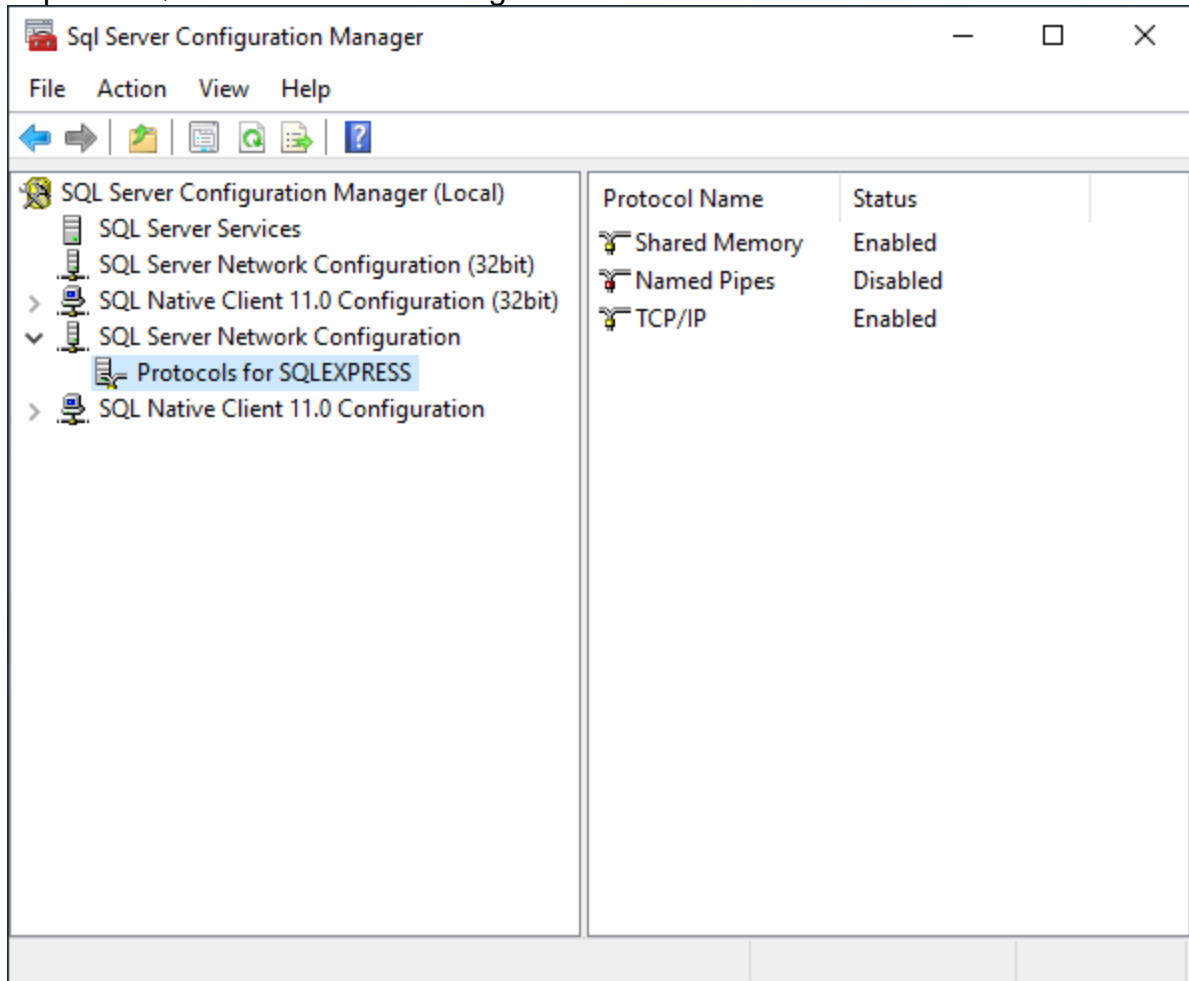
Running PageGate for the First Time

5

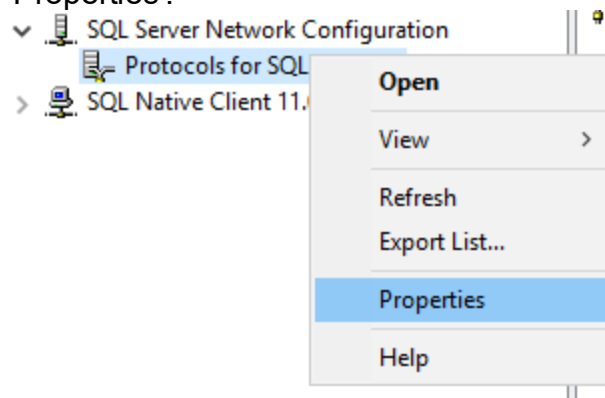
1) Open the SQL Server Configuration Manager



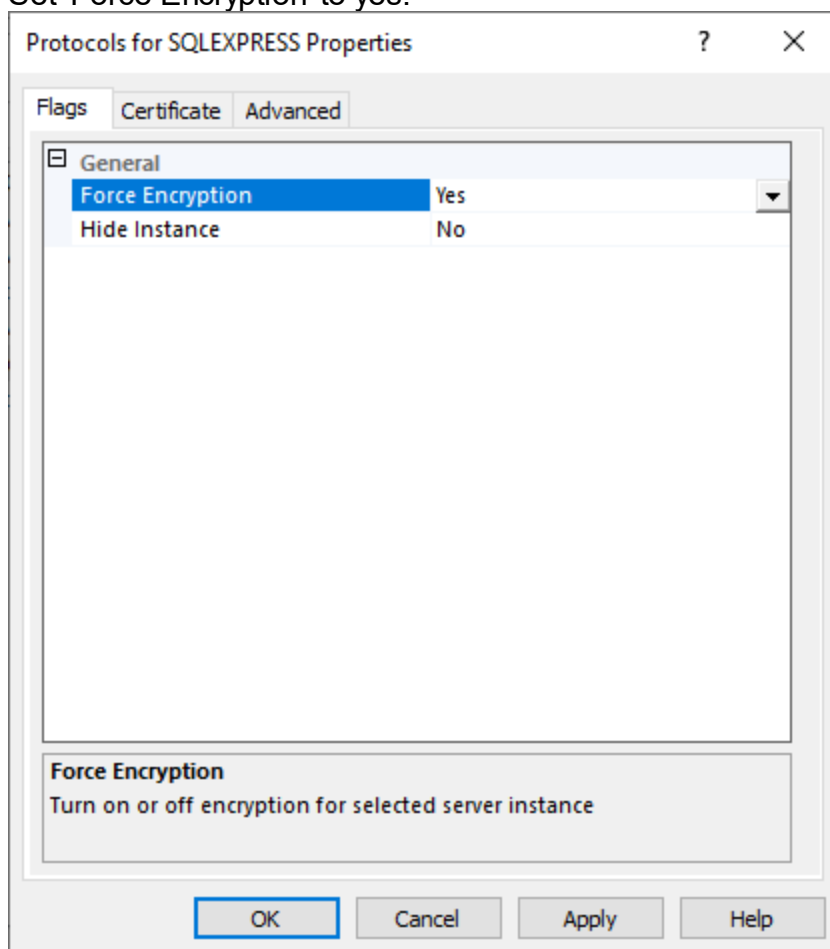
2) Expand SQL Server Network Configuration



3) Right click Protocols for SQL (SQLEXPRESS, MSSQLSERVER, etc) and select 'Properties'.



- 4) Set 'Force Encryption' to yes.



- 5) Click OK.

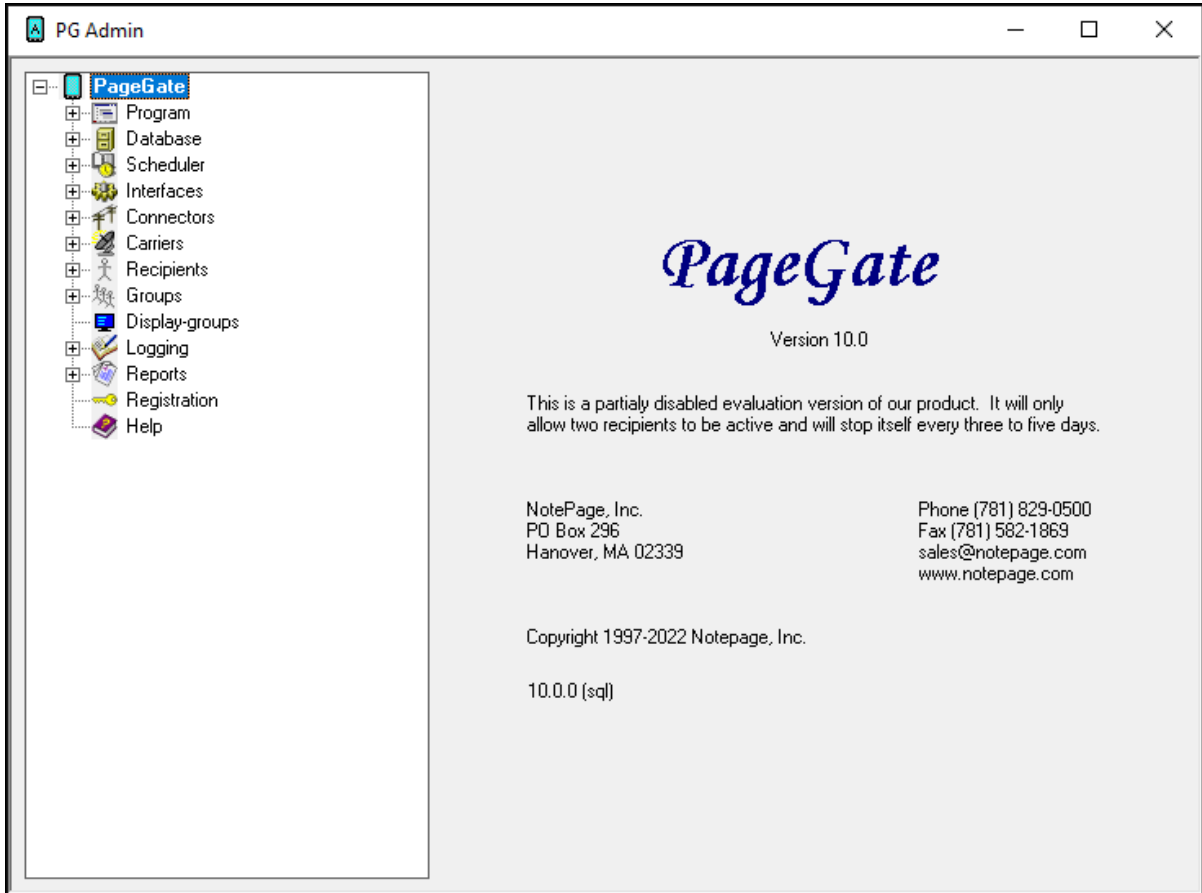
Step 5: Install PageGate and configure the basic settings

- 1) Go through the [steps to install the PageGate Server](#).
- 2) From your Start Menu or Desktop, right click on the shortcut for the PageGate Admin and select "Run as Administrator"
- 3) You should then be presented with the default setup screen. You can leave the values at their defaults or specify custom locations. Click Apply.
- 4) When it asks if you would like to create the Database, Archive and Log directories, say Yes.
- 5) You should then see a message indicating that a new database has been created and that PageGate must be restarted. Click OK.

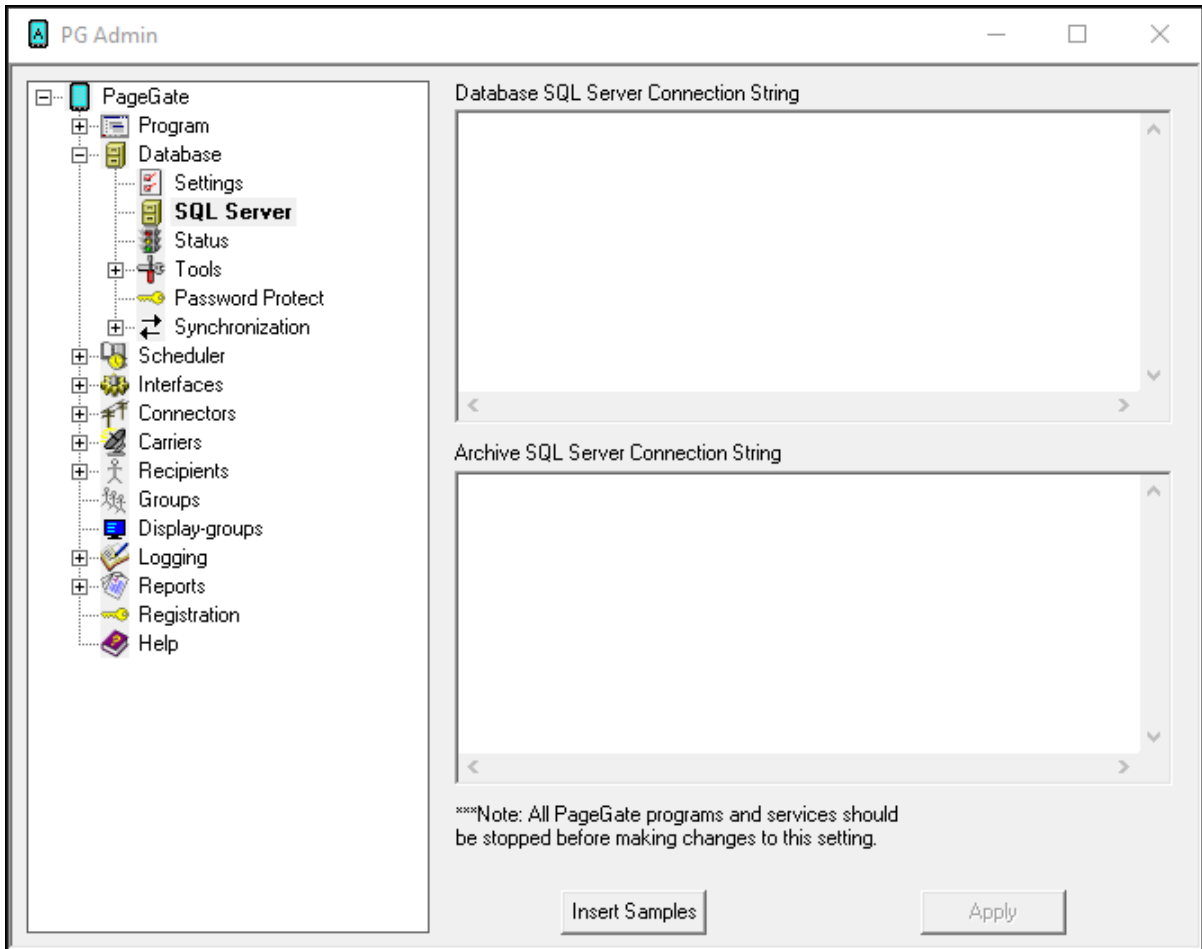
Running PageGate for the First Time

5

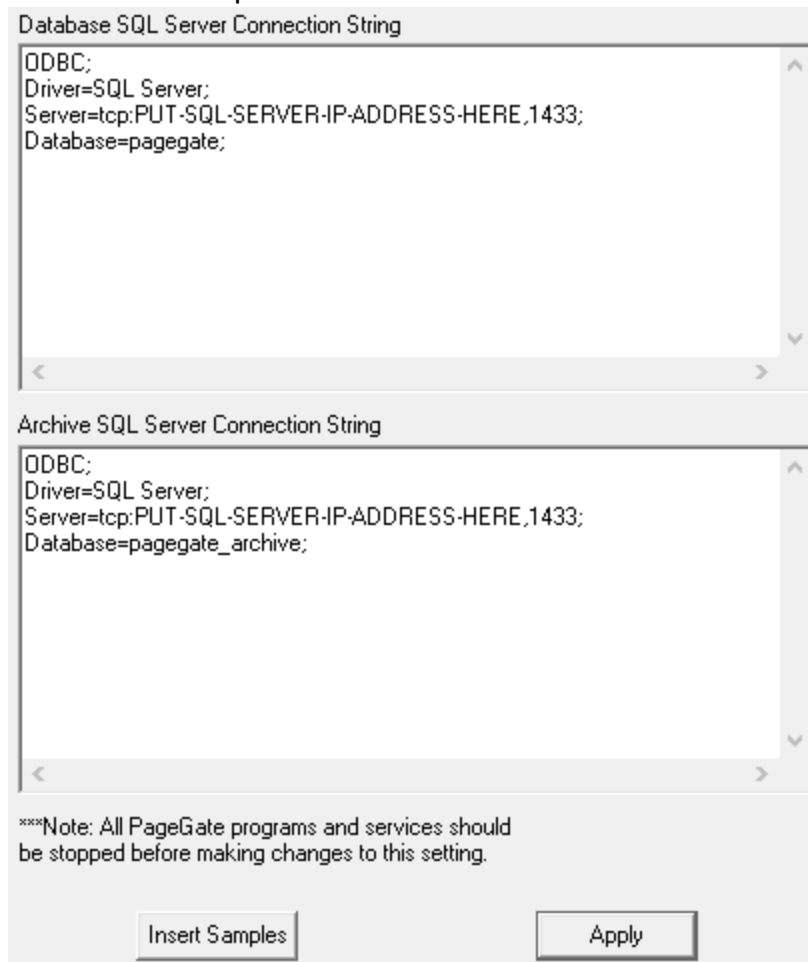
- 6) Right click on the shortcut for the PageGate Admin and select "Run as Administrator".



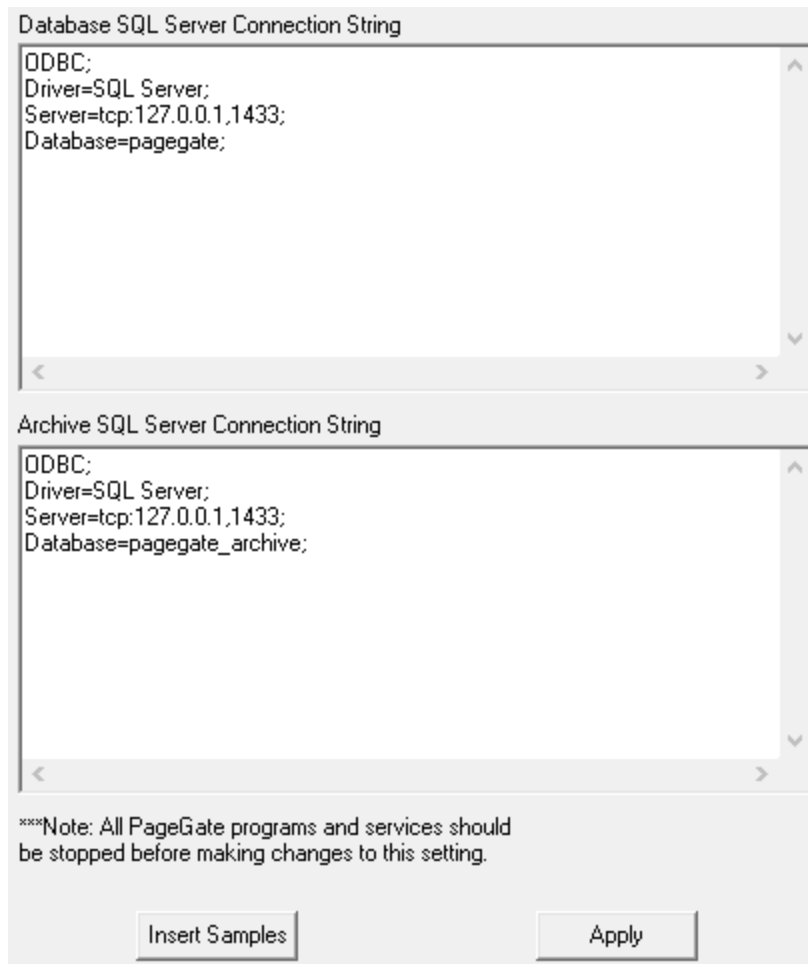
7) Go to Database - SQL Server



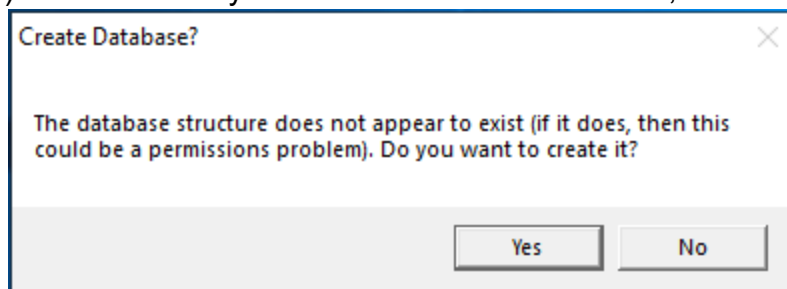
8) Click Insert Samples



9) Replace 'PUT-SQL-SERVER-IP-ADDRESS-HERE' with the IP address or machine name of the SQL Server.



- 10) Replace the Database name with the value you assigned in SQL. If your SQL database names are 'pagegate' and 'pagegate_archive', no changes are required.
- 11) Click Apply, then click OK on the next prompt.
- 12) Right click on the shortcut for 'PG Admin' and select 'Run as Administrator'.
- 13) When asked if you want to create the database, click Yes.

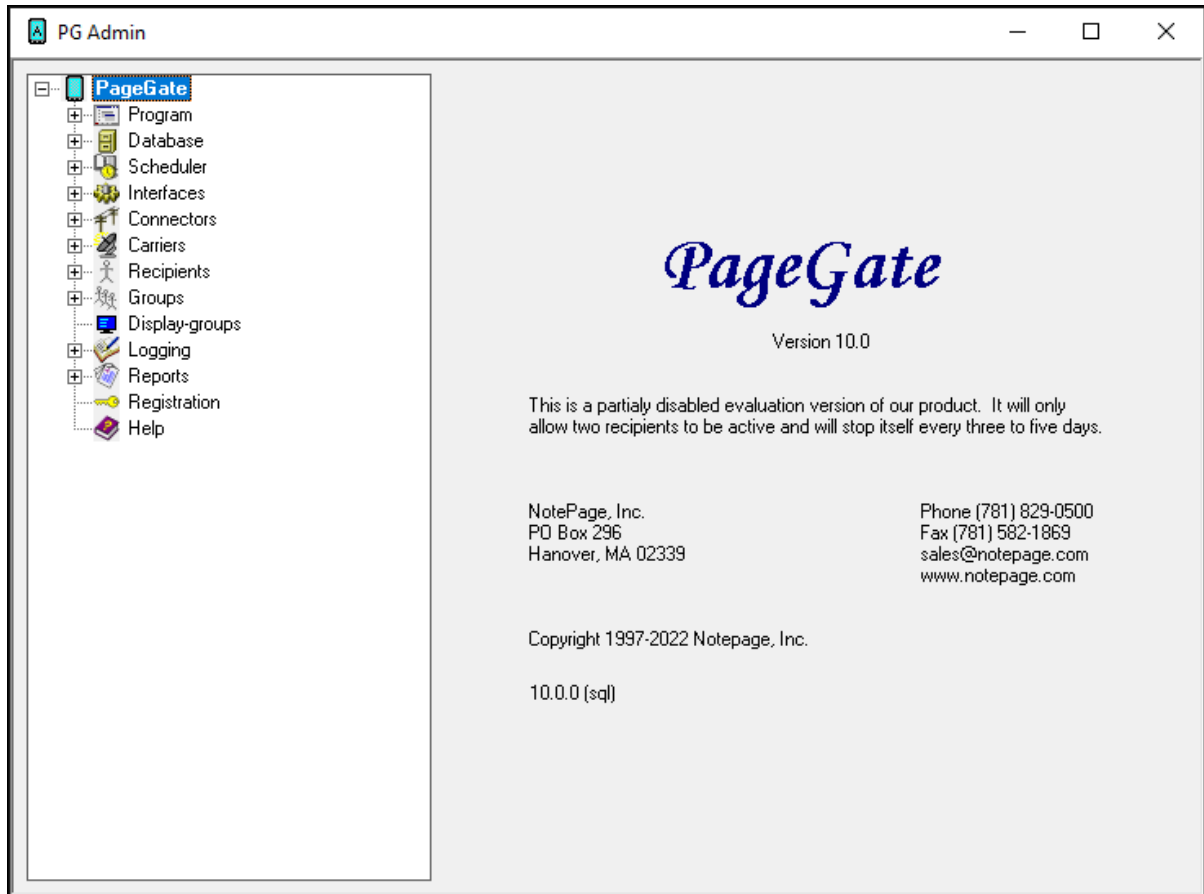


You can now begin configuring the [interfaces](#), [output methods](#) and [recipients](#) and [groups](#).

We also recommend [switching PageGate's run style](#) to PG Service or Windows service.

Switching PageGate's Run Style

1) Launch PG Admin with 'Run as Administrator'.



Running PageGate for the First Time

5

2) Go to Program - Settings.

Program Directory: C:\Program Files (x86)\NotePage\PageGate\

Database Directory: C:\PageGateData\Database\

Log Directory: C:\PageGateData\Logs\

Archive Directory: C:\PageGateData\Archive\

Backup Directory: C:\PageGateData\Database\Backup\

Run PageGate as:

- Application
- Windows Service
- PG Service

Run on this server:

<input checked="" type="checkbox"/> Scheduler	<input checked="" type="checkbox"/> GetAscii	<input checked="" type="checkbox"/> GetMail	
<input type="checkbox"/> GetSerial	<input type="checkbox"/> GetTap	<input checked="" type="checkbox"/> GetWeb	
<input checked="" type="checkbox"/> Connector1	<input type="checkbox"/> Connector2	<input type="checkbox"/> Connector3	<input type="checkbox"/> Connector4
<input type="checkbox"/> Connector5	<input type="checkbox"/> Connector6	<input type="checkbox"/> Connector7	<input type="checkbox"/> Connector8
<input type="checkbox"/> Connector9	<input type="checkbox"/> Connector10	<input type="checkbox"/> Connector11	<input type="checkbox"/> Connector12
<input type="checkbox"/> Connector13	<input type="checkbox"/> Connector14	<input type="checkbox"/> Connector15	<input type="checkbox"/> Connector16

Apply

3) In the "Run PageGate as" section, select either Windows Service or PG Service and click Apply.

Program Directory: C:\Program Files (x86)\NotePage\PageGate\

Database Directory: C:\PageGateData\Database\

Log Directory: C:\PageGateData\Logs\

Archive Directory: C:\PageGateData\Archive\

Backup Directory: C:\PageGateData\Database\Backup\

Run PageGate as:

- Application
- Windows Service
- PG Service

Run on this server:

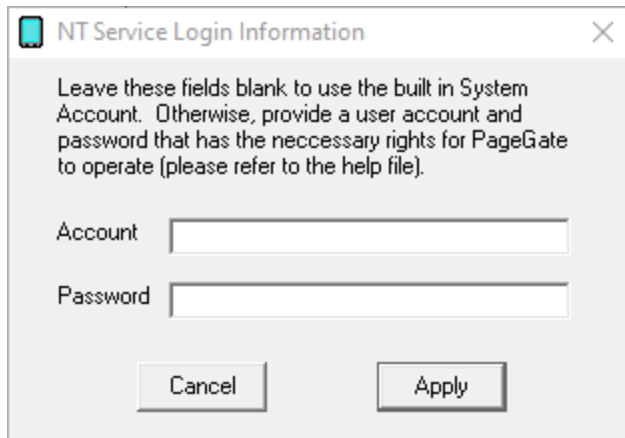
<input checked="" type="checkbox"/> Scheduler	<input checked="" type="checkbox"/> GetAscii	<input checked="" type="checkbox"/> GetMail	
<input type="checkbox"/> GetSerial	<input type="checkbox"/> GetTap	<input checked="" type="checkbox"/> GetWeb	
<input checked="" type="checkbox"/> Connector1	<input type="checkbox"/> Connector2	<input type="checkbox"/> Connector3	<input type="checkbox"/> Connector4
<input type="checkbox"/> Connector5	<input type="checkbox"/> Connector6	<input type="checkbox"/> Connector7	<input type="checkbox"/> Connector8
<input type="checkbox"/> Connector9	<input type="checkbox"/> Connector10	<input type="checkbox"/> Connector11	<input type="checkbox"/> Connector12
<input type="checkbox"/> Connector13	<input type="checkbox"/> Connector14	<input type="checkbox"/> Connector15	<input type="checkbox"/> Connector16

Apply

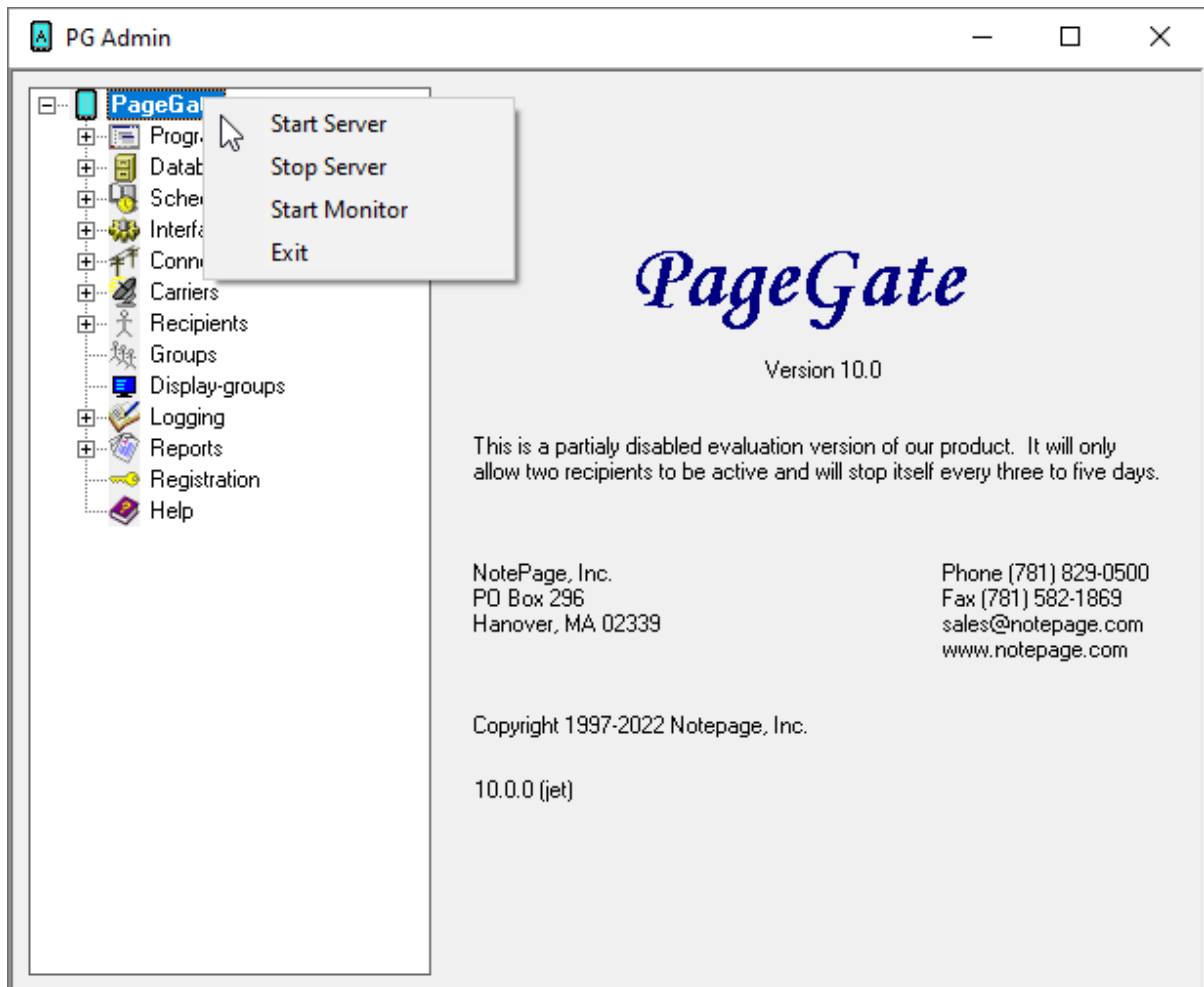
Running PageGate for the First Time

5

4) When prompted for credentials, leave them blank and click Apply.



5) When "Switching Run Styles" goes away, right click on the 'PageGate' at the top of the left hand menu tree.



6) Select 'Start Server'.

Running PageGate for the First Time

Starting and Stopping PageGate

After PageGate is started, icon of a smartphone will appear in the Windows system tray on the bottom right-hand corner of the screen (next to the clock). This icon is used to stop PageGate, start the PG Admin program, or start the PG Monitor program. Clicking on the pager with the left mouse button will display a menu with these options.

If PageGate is running as an application, the first menu option is the Restore function and will simply remove the pager icon from the system tray, and open a window on your desktop. Use the minimize button to return the icon to the system tray and remove the PageGate window. Quitting the PageGate window by using the 'X' control or by using the 'Quit PageGate' button will stop the entire PageGate Server application.

If PageGate is running as a Windows System service, the first menu option will be 'Start' instead of 'Restore'. This option will start the PageGate system service if it is not already running.

The second menu item is 'PG Admin'. Choosing this item will start PageGate's Administration program (if it is not already running).

The third menu item is 'PG Monitor'. Choosing this item will start PageGate's real-time monitoring program.

If PageGate is running as an Application, the last option is 'Quit'. Selecting 'Quit' will stop the entire PageGate application. If PG Admin or PG Monitor are running, they will need to be exited individually.

If PageGate is running as a Windows System service, the last option is 'Stop'. This will stop the PageGate Server service from running. If the PG Admin or PG Monitor programs are running, they will need to be quit out of individually.

If PageGate has been setup to run as Windows Service, it will start automatically each time the computer is restarted. Otherwise, you must start PageGate manually from the Windows start menu. (or add it to the computer's startup group).

Monitoring PageGate

Please see the "[Keeping an eye on PageGate](#)" section of the documentation.

Example Configuration using the PageGate GUI Client

Step 1: Install PageGate

- 1) Go to the system that's going to host the database and program for PageGate (I'll refer to it as the PageGate Server).

- 2) Browse to our website, <https://www.notepage.net>
- 3) Click the Download button under the PageGate section.
- 4) Save the pgsetup.exe file somewhere convenient, like the Desktop.
- 5) Once the file is downloaded, right click on it and select "Run as Administrator"
- 6) You should then be presented with four installation options. Select: Install PageGate Server
- 7) Click next until the installation finishes.
- 8) From your Start Menu or Desktop, right click on the shortcut for the PageGate Admin and select "Run as Administrator"
- 9) You should then be presented with the default setup screen. Click Apply.
- 10) When it asks if you would like to create the Database, Archive and Log directories, say Yes.
- 11) You should then see a message indicating that a new database has been created and that PageGate must be restarted. Click OK.
- 12) Right click on the shortcut for the PageGate Admin and select "Run as Administrator".
- 13) Go to the Program - Settings section of the Admin.
- 14) Under 'Run PageGate As', select: Windows Service
- 15) Click Apply.
- 16) You should then be prompted for credentials. Leave both boxes blank and click Apply.
- 17) When "Switching Run Styles" goes away, open the Windows Service list.
- 18) Find and start the PageGate service.

Step 2: Configure PageGate to send messages

Now that we have PageGate installed and the server engine running, all that remains is to configure the program to send messages.

[Click here](#) for information on how to configure PageGate to send messages.

Step 3: Share the PageGate server's database

To get the PageGate GUI Client installed and functioning on workstations, there's a little networking setup that has to be done in advance on the server. All you'll need to do is share the PageGate Database directory (on the PageGate Server) on the network and make a note of the UNC/Networking path to the shared folder.

Step 4: Install PageGate's GUI Client

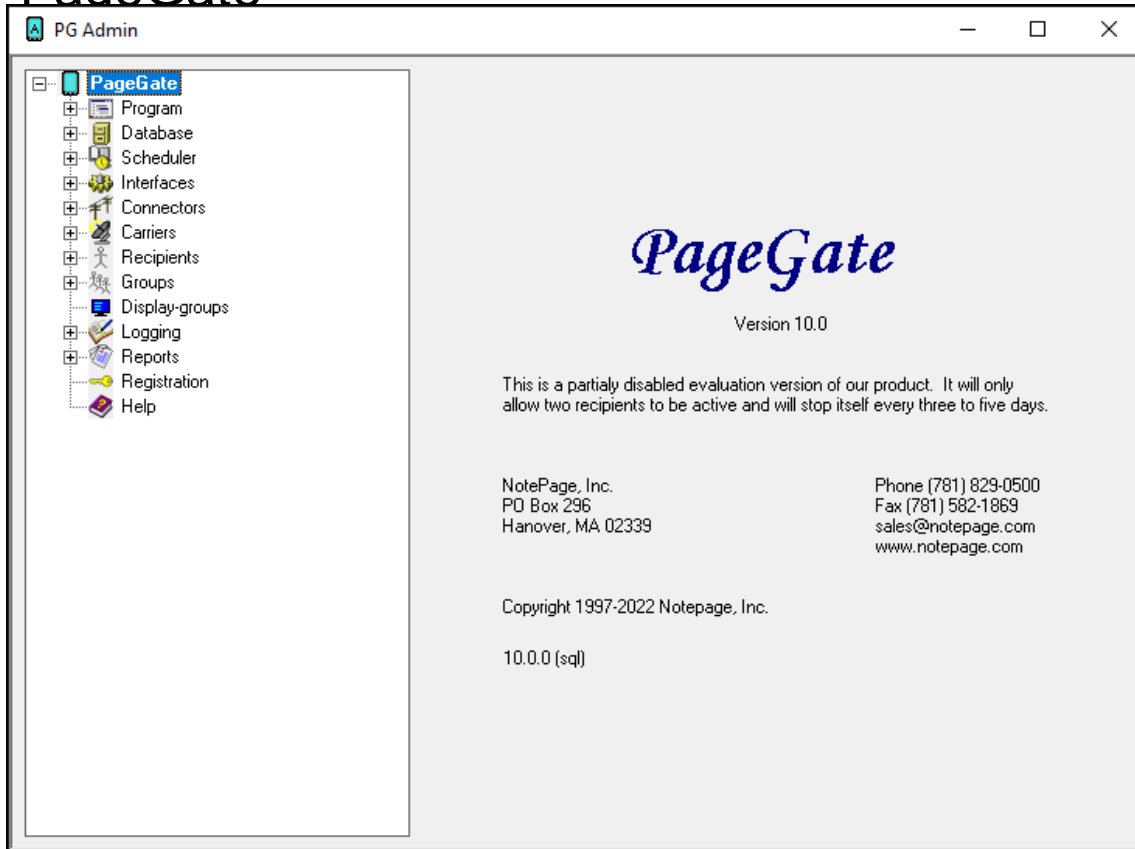
Once you have the PageGate Server's database shared, here's how to install the PageGate Client on a workstation:

- 1) Go to a system that's going to run the PageGate Client.
- 2) Browse to our website, <https://www.notepage.net>
- 3) Click the Download button under the PageGate section.
- 4) Save the pgsetup.exe file somewhere convenient, like the Desktop.
- 5) Once the file is downloaded, right click on it and select "Run as Administrator"
- 6) You should then be presented with four installation options. Select: Install PageGate GUI Client
- 7) Click Next until the installation finishes.
- 8) When you run the Client for the first time, it's going to ask you for the server's database path. This is the network path to the PageGate Server's network shared database directory (example: \\servername\directory\)
- 9) Click Apply.

The PageGate Administrator (PG Admin) is used for configuring and controlling all aspects of the PageGate Server. The PageGate Admin takes an object-oriented approach to administering the server; most of its functions and settings can be accessed by manipulating the objects in its object-tree. Many of the functions of PG Admin are only accessible by right-clicking on these objects.

PageGate Admin: How to Configure PageGate

6



The Admin is automatically installed on the PageGate Server. Workstations that need to administrate your PageGate server(s) should have the [Workstation PG Admin](#) installed on them.

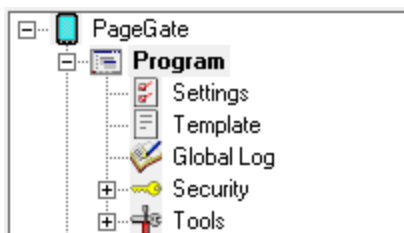
When running PG Admin for the first time, you will need to use the 'Run as Administrator' option.

To run PG Admin, there should be a shortcut on the desktop and in the start menu. If the PageGate server is running, you can also click on the PageGate icon in your system tray and select 'PG Admin'.

Program

The Program section contains settings and information that relate to the global operations of the PageGate Server.

PageGate Admin: How to Configure PageGate



- [Settings](#)

This section contains the PageGate server's configuration, including which APIs to run and whether PageGate should run as an application or Windows service.

- [Global Log](#)

This section contains the global log of activity recorded by the PageGate server, including server start and stop events as well as a record of traffic processed.

- [Template](#)

This section contains the PageGate server's global template.

- [Security](#)

This section contains the PageGate server's security settings, users and their permissions.

- [Tools](#)

This section contains a series of tools to test internet and hardware functionality.

Right clicking on PageGate will bring up a context menu with four options.

- [Start Server](#)

Selecting this option will start the PageGate server, if it's stopped.

- [Stop Server](#)

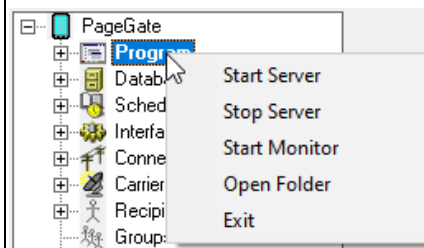
Selecting this option will stop the PageGate server, if it's started.

- [Start Monitor](#)

Selecting this option will launch the PageGate Monitor application.

- [Exit](#)

Selecting this option will close the PageGate Admin.



PageGate Admin: How to Configure

PageGate Program Settings

The screenshot shows the 'PageGate Program Settings' dialog box. It contains several sections:

- Directories:** Five text boxes with browse buttons (three dots) for:
 - Program Directory: C:\Program Files (x86)\NotePage\PageGate\
 - Database Directory: C:\PageGateData\Database\
 - Log Directory: C:\PageGateData\Logs\
 - Archive Directory: C:\PageGateData\Archive\
 - Backup Directory: C:\PageGateData\Database\Backup\
- Run PageGate as:** Three radio buttons:
 - Application
 - Windows Service
 - PG Service
- Run on this server:** A grid of checkboxes:
 - Scheduler, GetAscii, GetMail
 - GetSerial, GetTap, GetWeb
 - Connector1, Connector2, Connector3, Connector4
 - Connector5, Connector6, Connector7, Connector8
 - Connector9, Connector10, Connector11, Connector12
 - Connector13, Connector14, Connector15, Connector16
- Buttons:** An 'Apply' button is located at the bottom center.

This section can be broken down in to three parts:

- Where to store the database, logs and archives.

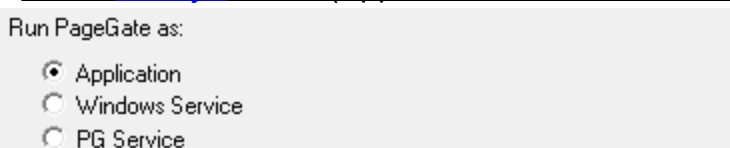
This is a close-up of the directory settings section from the screenshot above, showing the same five text boxes with their respective paths and browse buttons.

Setting	Function
Program Directory	The Program Directory MUST point to the directory where PageGate is installed and should not be modified after the installation.
Database Directory	This directory specifies the PageGate Server's database location. The database contains ALL pertinent information for the PageGate Server and must be shared on the network for the GUI Client and Additional Admin to reference.
Log Directory	This directory specifies where PageGate should keep its log and security log files.

PageGate

Archive Directory	When the option for PageGate to archive messages is enabled, all archived files will be stored in this directory.
Backup Directory	This directories tells PageGate where to keep backup information.

- Which [run style](#) to use (Application, Windows Service or PG Service)



Run PageGate as:

Application

Windows Service

PG Service

We **strongly recommend** running PageGate as a [Windows Service](#) or [PG Service](#).

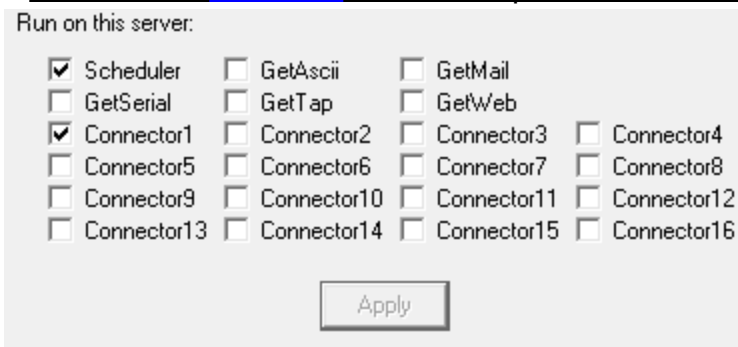
The '[Windows Service](#)' run style tells PageGate to create a service for each component of the program.

The '[PG service](#)' run style tells PageGate to create a single, master service that all elements of the program are run from.

Server 2016, 2019 and 2022: Use the '[PG Service](#)' run style.

Desktop and Previous Server Versions: Use the '[Windows Service](#)' or '[PG Service](#)' run styles.

- Which which [Interfaces](#) to use for input and how many [Connectors](#) to use for output



Run on this server:

<input checked="" type="checkbox"/> Scheduler	<input type="checkbox"/> GetAscii	<input type="checkbox"/> GetMail	
<input type="checkbox"/> GetSerial	<input type="checkbox"/> GetTap	<input type="checkbox"/> GetWeb	
<input checked="" type="checkbox"/> Connector1	<input type="checkbox"/> Connector2	<input type="checkbox"/> Connector3	<input type="checkbox"/> Connector4
<input type="checkbox"/> Connector5	<input type="checkbox"/> Connector6	<input type="checkbox"/> Connector7	<input type="checkbox"/> Connector8
<input type="checkbox"/> Connector9	<input type="checkbox"/> Connector10	<input type="checkbox"/> Connector11	<input type="checkbox"/> Connector12
<input type="checkbox"/> Connector13	<input type="checkbox"/> Connector14	<input type="checkbox"/> Connector15	<input type="checkbox"/> Connector16

Apply

Please see the [Interfaces](#) and [Connectors](#) sections for more information.

Run Styles

PageGate can be configured to run in three different styles: Application, Windows Service and PG Service.

We **strongly recommend** using the Windows Service or PG Service run styles.

Application

PageGate Admin: How to Configure PageGate

6

In this mode, PageGate is started and stopped just like any other program and can be automatically run at startup by putting a shortcut to it in the Windows "Startup" group. The disadvantage to this run style, is that if the computer is configured to show a login screen at startup, PageGate cannot automatically start until someone logs onto the computer.

Windows Service or PG Service

The 'Windows Service' run style tells PageGate to create a service for each component of the program.

The 'PG service' run style tells PageGate to create a single, master service that all elements of the program are run from.

Server 2016, 2019 and 2022: Use the 'PG Service' run style.

Desktop and Previous Server Versions: Use the 'Windows Service' or 'PG Service' run styles.

- 1) Right click on the shortcut for the PageGate Admin and select "Run as Administrator".
- 2) Go to the Program - Settings section of the Admin.
- 3) Under 'Run PageGate As', select: Windows Service or PG Service
- 4) Click Apply.
- 5) You should then be prompted for credentials. *See notes below.
- 6) When "Switching Run Styles" goes away, open the Windows Service list.
- 7) Find and start the PageGate service.

*Windows Credentials and PageGate

This is where things might get a little complicated because this part has more to do with how your network and/or domain are configured than anything native to PageGate; you're interacting with the Windows System Services. Having said that, you can typically use the Windows System Account to run PageGate's services. To do so, when you're prompted for credentials in step 5, leave the username and password fields blank and click on Apply again.

If leaving the username and password fields blank returns an "Invalid account or password" error, that means that the Windows System Account has been disabled and you must provide an appropriate username and password that has the "Create service" and "Log on as service" policies.

However, the Windows System Account typically has no network privileges and cannot access cross-network information.

PageGate Admin: How to Configure

6

PageGate

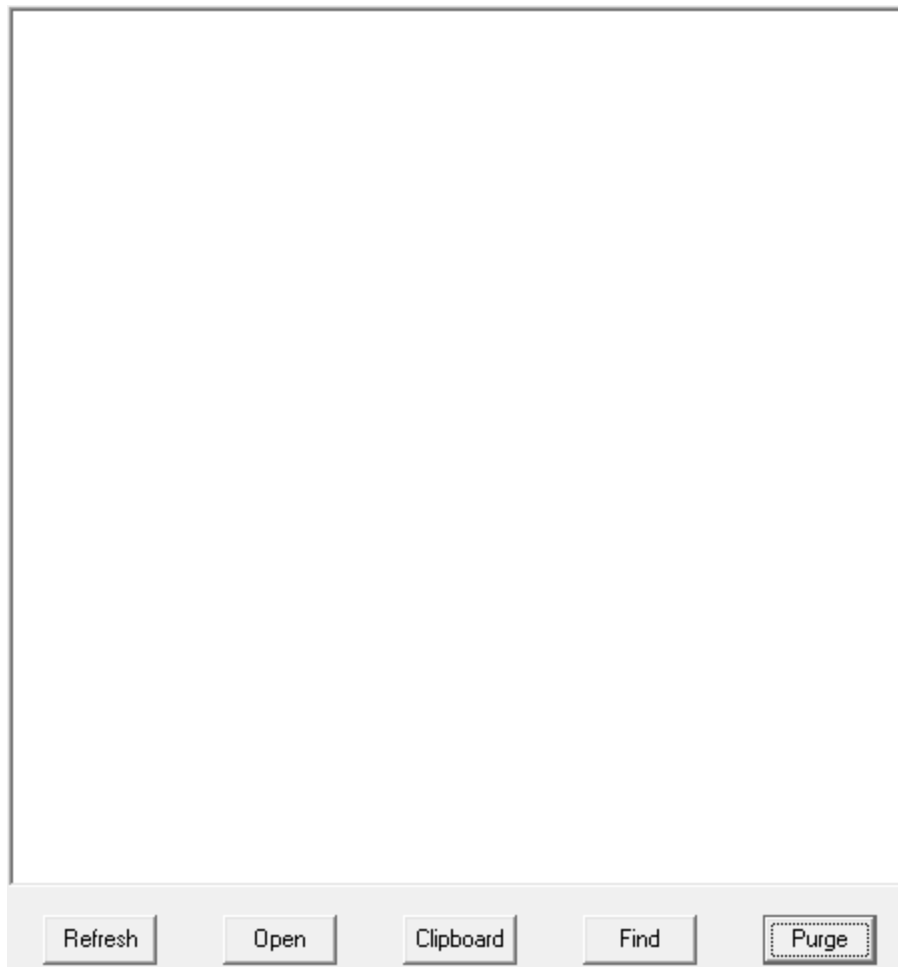
If you have any component of the PageGate server referencing a network resource, like a file path by GetASCII or GetWeb APIs, the corresponding service must have sufficient rights to access those resources (in addition to the 'run as service' right).

In our experience, the easiest thing to do when you want to run PageGate as a service but need to have an interface or connector reference a network path is to have the system services created with the Windows System Account. Then, after the services exist, go in to the properties of the service that corresponds to the API or Connector and assign the service account under the Log On tab of the service.

When using the PG Service run style, you will only have a single, master PageGate service that will need a service account assigned to it.

Global Log

If logging is enabled, this window displays a summary of the events generated by PageGate, including all messages scheduled and sent.

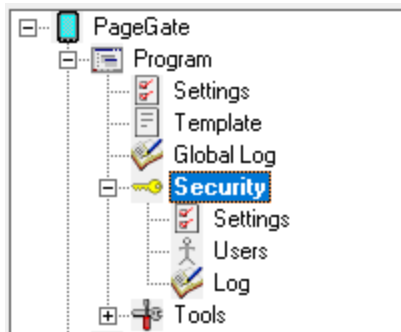


PageGate

Refresh	Displays the most current entries in the global log.
Print	Sends a copy of everything listed in the Global Log to a printer.
Clipboard	Copies the contents of the Global Log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Deletes the entirety of the Global Log.
Purge	Allows you to search for a variable (such as a carrier or recipient name).

Security

The security section of the PageGate Admin allows you to configure the program's security system, which can be used to restrict access to both PG Admin and the GUI Client.



- [Settings](#) allows you to enable or disable security restrictions to the Admin or GUI Client.
- [Users](#) allows you to create and manage the list of users who have access to the programs and what level of control they have.
- [Log](#) displays which user has performed what action(s) at any given date and time. It records each instance of security permissions being used.

Settings

Note: It is entirely possible lock yourself out of the PageGate Admin if you enable the PG Admin security before configuring any [security users](#). Please configure at least one [master administrative user](#) before enabling any security in the program.

PageGate Admin: How to Configure PageGate

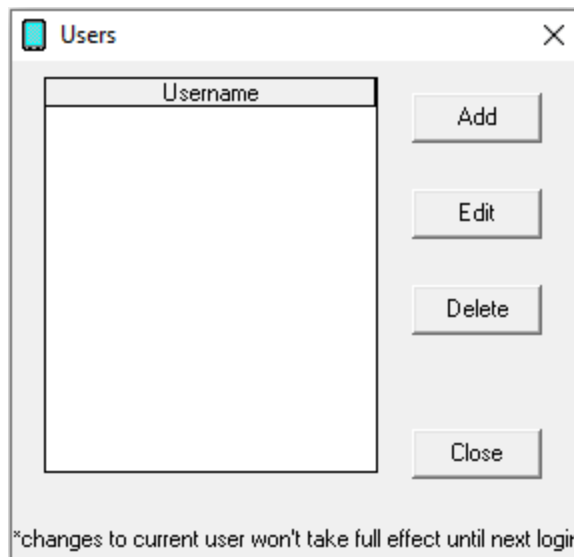
6



Enable PG Admin Security	Restricts access to the PG Admin to ONLY those users given rights in the Users section. If this option is not enabled, all users will be able to access all functions of the PG Admin.
Enable GUI Client Security	Restricts access to the GUI Client to ONLY those users given rights in the Users section. If this option is not enabled, all users will be able to access all functions of the GUI Client.

Users

The users portion of the Security Settings allows you to create and maintain a list of users that have permission to access certain portions of either the PageGate Admin or the GUI Client.



PageGate Admin: How to Configure PageGate

6

Add	This allows you to create a new security user profile. Click here for more information.
Edit	This allows you to edit an existing security user profile.
Delete	This allows you to remove an existing security profile.
Close	This closes the Users dialog.

Adding a Security User

New User [Close]

Username

Use Windows Login Credentials

Password

Alias [Dropdown]

Display-group [Dropdown]

Sender

Station

Presets File

Rights

- Administrator
- Interfaces
- Connectors
- Carriers
- All Recipients
- Own Recipients
- All Groups
- Own Groups
- All Display-groups
- Own Display-groups
- Reports
- Client Administrator
- Client

Cancel Apply

When creating a user to have security rights, there are several settings that need to be considered:

PageGate

Username	This field is where you assign a unique login name to a user. When a user attempts to access a security enabled PG Admin or GUI Client, they will be prompted for this username.
Use Windows Login Credentials	When this option is checked, both the GUI Client and PG Admin will allow a user logged into windows with this name to access the program. For example, if you create a user named "Johnny" and a workstation is currently logged in as the Windows user "Johnny", it will grant access to the program.
Password	This field is where you assign a unique password to a user. When a user attempts to access a security enabled PG Admin or GUI Client, they will be prompted for this password.
Alias	This field allows you to set a security alias for this user.
Display-group	This field over-rides the Display-group setting in the settings of the GUI Client for this specific user. For more information, please refer to the GUI Client Settings section of the help file.
Sender	This field over-rides the Sender setting in the GUI Client for this specific user. For more information, please refer to the GUI Client Settings section of the help file.
Station	This field over-rides the Station Number of the GUI Client for this specific user. For more information, please refer to the GUI Client Settings section of the help file.
Presets File	This field over-rides the Preset Msgs File setting in the settings of the GUI Client for this specific user. For more information, please refer to the GUI Client Settings section of the help file.

The following rights can be assigned to a user:

PageGate Admin: How to Configure PageGate

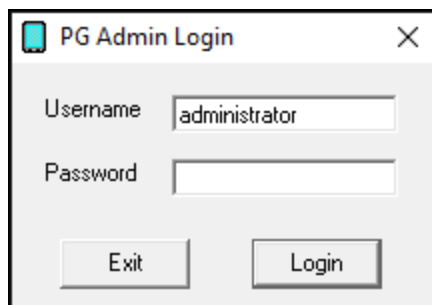
6

Administrator	This will make this user an administrator capable of adding, editing and removing all recipients, carriers, groups and display groups as well. This also grants this user the ability to administrate the Interfaces, Connectors and other global program values.
Interfaces	Grants this user permission to access and modify settings in the Interfaces section of the Admin.
Connectors	Grants this user permission to access and modify settings in the Connectors section of the Admin.
Carriers	Grants this user permission to access and modify settings in the Carriers section of the Admin.
All Recipients	Grants this user permission to access and modify settings for all recipients in the Recipients section of the Admin.
Own Recipients	Grants this user permission to access and modify settings in the Recipients section but only for those recipients that this user, or this user's alias , has been marked Manager .
All Groups	Grants this user permission to access and modify settings for all groups in the Groups section of the Admin.
Own Groups	Grants this user permission to access and modify settings in the Groups section but only for those groups that this user, or this user's alias , has been marked Manager .
All Display-Groups	Grants this user permission to access and modify settings for all display groups in the Display Groups section of the Admin.
Own Display-Groups	Grants this user permission to access and modify settings in the Display Groups section but only for those display groups that this user, or this user's alias , has been marked Manager .
Reports	Grants this user permission to access the Reports section of the Admin.
Client Administrator	This will make this user an administrator for the PageGate GUI Client but will not grant them access to the Admin or its functions.
Client	Grants this user permission to access and run the PageGate GUI Client.

Logging In

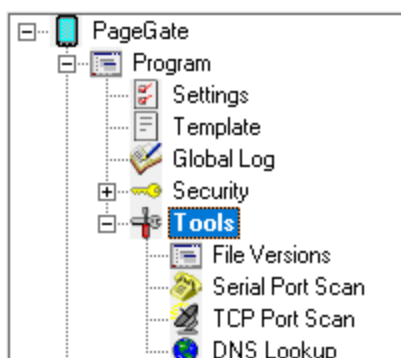
Note: Not every user is granted administrative privileges. If certain functions are unavailable, the user you have logged in with may not have permission to access to them.

PageGate Admin: How to Configure PageGate



When logging in to the PageGate Admin or PageGate Client, you will receive the login prompt (shown above). Simply type in the username and password that has been assigned, then click Login.

Tools



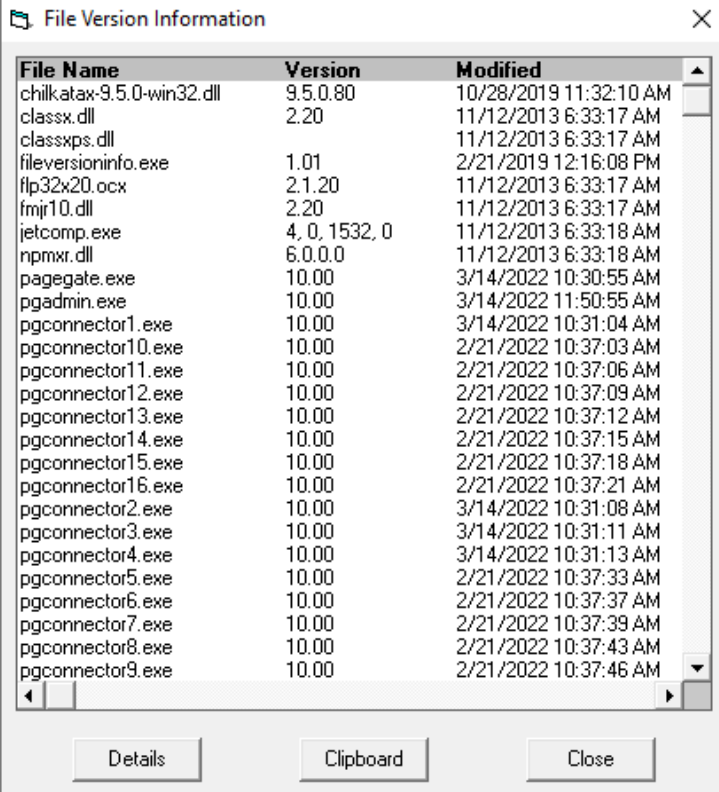
- [File Versions](#) provides a quick reference for the specific file version information for your installed copy of PageGate.
- [Serial Port Scan](#) runs a port scan to determine if you have any available COM ports in the operating system.
- [TCP Port Scan](#) runs a test on a range of TCP and UDP ports related to message delivery and will report if they are open or not.
- [DNS Lookup](#) performs a series of DNS lookups, including A, MX and SPF record querying.

File Versions

Clicking on "File Versions" will display the following

dialog:


PageGate Admin: How to Configure PageGate



File Name	Version	Modified
chilkatx-9.5.0-win32.dll	9.5.0.80	10/28/2019 11:32:10 AM
classx.dll	2.20	11/12/2013 6:33:17 AM
classxps.dll		11/12/2013 6:33:17 AM
fileversioninfo.exe	1.01	2/21/2019 12:16:08 PM
flp32x20.ocx	2.1.20	11/12/2013 6:33:17 AM
fmjr10.dll	2.20	11/12/2013 6:33:17 AM
jetcomp.exe	4, 0, 1532, 0	11/12/2013 6:33:18 AM
npmxr.dll	6.0.0.0	11/12/2013 6:33:18 AM
pagegate.exe	10.00	3/14/2022 10:30:55 AM
pgadmin.exe	10.00	3/14/2022 11:50:55 AM
pgconnector1.exe	10.00	3/14/2022 10:31:04 AM
pgconnector10.exe	10.00	2/21/2022 10:37:03 AM
pgconnector11.exe	10.00	2/21/2022 10:37:06 AM
pgconnector12.exe	10.00	2/21/2022 10:37:09 AM
pgconnector13.exe	10.00	2/21/2022 10:37:12 AM
pgconnector14.exe	10.00	2/21/2022 10:37:15 AM
pgconnector15.exe	10.00	2/21/2022 10:37:18 AM
pgconnector16.exe	10.00	2/21/2022 10:37:21 AM
pgconnector2.exe	10.00	3/14/2022 10:31:08 AM
pgconnector3.exe	10.00	3/14/2022 10:31:11 AM
pgconnector4.exe	10.00	3/14/2022 10:31:13 AM
pgconnector5.exe	10.00	2/21/2022 10:37:33 AM
pgconnector6.exe	10.00	2/21/2022 10:37:37 AM
pgconnector7.exe	10.00	2/21/2022 10:37:39 AM
pgconnector8.exe	10.00	2/21/2022 10:37:43 AM
pgconnector9.exe	10.00	2/21/2022 10:37:46 AM

This is a list of every file, and its specific build version, in the PageGate program directory

Serial Port Scan



COM 1 COM 9
 COM 2 COM 10
 COM 3 COM 11
 COM 4 COM 12
 COM 5 COM 13
 COM 6 COM 14
 COM 7 COM 15
 COM 8 COM 16

The serial port scan tool allows PageGate to scan available communication (COM) ports. Any COM port found to have a modem present will have a check placed next to it when the scan completes.

PageGate Admin: How to Configure

6

PageGate Port Scan

PageGate uses many internet based protocols. This Port Scan tool allows PageGate to determine whether the ports to use these protocols are available or closed.

Port	Corresponding Protocol
25	SMT P/E mail
465	SMT P/E mail with SSL/STARTLS
587	SMT P/E mail with STARTLS
80	WCTP
110	POP
444	SNPP
53	DNS

DNS Lookup

DNS Lookup provides a utility that will attempt to perform a series of DNS lookups, including A, MX and SPF, on the domain specified.

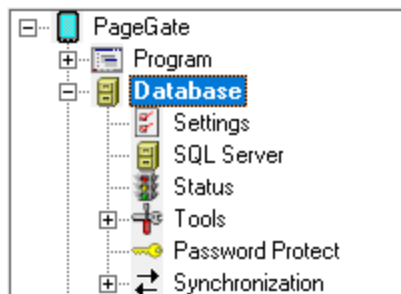
PageGate Admin: How to Configure PageGate

This is incredibly useful when [configuring PageGate to send SMTP/Email](#).



Database

The Database section is where any database related settings and functions, like how much information to store in the database or synchronizing multiple PageGate servers, are configured.



- [Settings](#)

This section controls message retention and archiving options.

- [Status](#)

This section displays relevant information about the database, including file size.

- [Tools](#)

This section allows you to import and export recipients as well as rebuild the database in the event of corruption.

PageGate Admin: How to Configure PageGate

- [Password Protect](#)

This section allows you to password protect the PageGate databases.

- [Synchronization](#)

This section allows you to configure multiple PageGate servers to synchronize their data with each other.

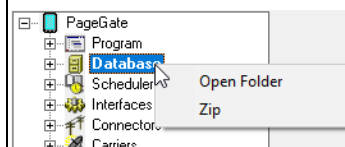
Right clicking on Database will provide a context menu that provides two options:

- [Open Folder](#)

Selecting this option will open a Windows File Explorer pointed to the PageGate database directory.

- [Zip](#)

This will create a compressed file named pagegate.mdb.zip output in the PageGate database directory and is useful in the event that you need to email your database to Tech Support or for backup purposes.



Settings

Purge After (days)

Prompt for backup

Include messages in backup

Recipient Extra Field Labels

Field1 Label

Field2 Label

Field3 Label

Field4 Label

Move messages to archive

Move After (days)

Setting	Function
Purge After (days)	This field sets the number of days that PageGate will keep old messages in the messages table. After X days, where X is the number you enter, PageGate will purge old messages from the database.

PageGate

Run Purge Now	Messages are normally purged from the database during the midnight cleanup routine. Messages can be manually purged from the database by clicking this button.
Prompt for Backup	This field may be set to: Never, daily, weekly, monthly, quarterly, Bi-annually and yearly If enabled, PG Admin will prompt users to back up the database at the specified time interval.
Include messages in backup	If enabled, this option includes the 'Messages' table of the database in the backup routine.
Field 1, 2, 3, and 4 Label	Four extra pieces of information can be stored with each recipient and group. This is where you can set the labels that are shown with each piece of information.
Move messages to archive	Checking this option will allow PageGate to move "purged" messages to the Archive Directory .
Move After (days)	This determines the number of days that messages are moved to the Archive Directory .

Status

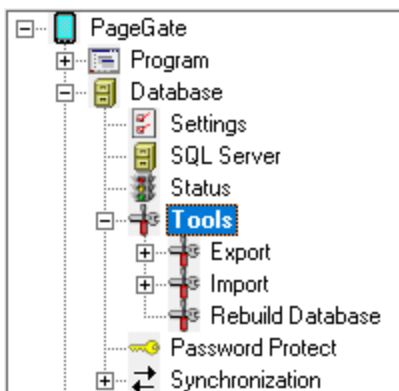
Database File Size (Kb)	<input type="text" value="512"/>
Carrier Records	<input type="text" value="1"/>
Recipient Records	<input type="text" value="1"/>
Group Records	<input type="text" value="0"/>
Display-group Records	<input type="text" value="0"/>
Message Records	<input type="text" value="1"/>
Local Messages	<input type="text" value="0"/>
Scheduled Messages	<input type="text" value="1"/>
Pending Messages	<input type="text" value="0"/>

PageGate Setting

Setting	Function
Database File Size (Kb)	Displays the current file size of the database in kilobytes.
Carrier Records	Displays the current number of carrier records in the database.
Recipient Records	Displays the current number of recipient records in the database.
Group Records	Displays the current number of group records in the database.
Display-group Records	Displays the current number of Display-group records in the database.
Message Records	Displays the current number of messages in the messages table.
Pending Messages	Displays the current number of messages waiting to be processed in the messages table.

Tools

This section allows you to import and export information as well as rebuild the database in the event of corruption.



- [Export](#)

This option is used for exporting information from PageGate in a CSV format.

- [Import](#)

This option is used for pulling information in to PageGate from a CSV format.

- [File Formats](#)

This section goes over the specific format of PageGate's export files and the required structure for importing. These file structures are also used by PageGate's [Synchronization](#)

PageGate Admin: How to Configure PageGate

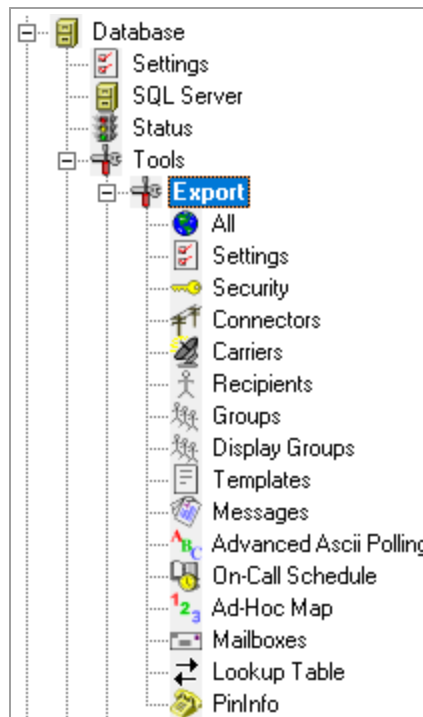
[system.](#)

- [Rebuild Database](#)

This option is used for repairing databases that may have errors in them.

Export

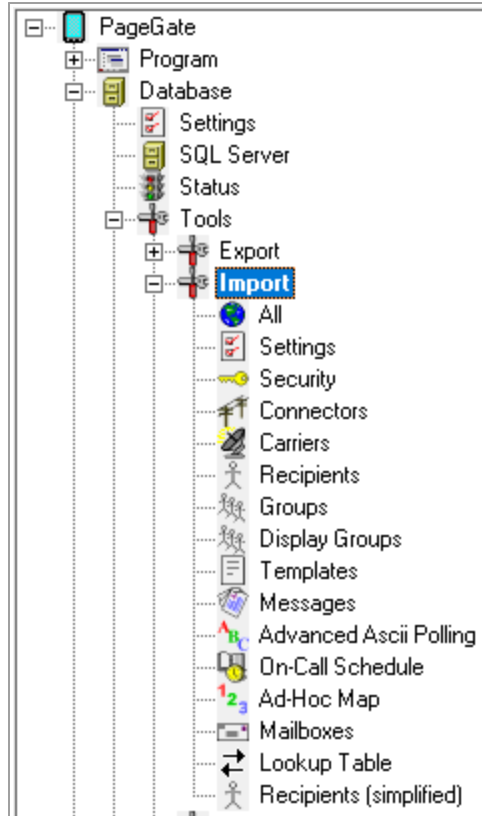
This section allows you to export information from the PageGate database.



PageGate

Menu Item	Function
All	This exports everything at once to a specified directory.
Settings	Exports the existing global settings for PageGate, including registration and Program Settings .
Security	Exports the existing Security Settings and Users .
Connectors	Exports the existing Connectors settings.
Carriers	Exports all carriers and their settings.
Recipients	Exports all recipients
Groups	Exports all groups and group members
Display Groups	Exports all display groups and display group members
Templates	Exports all Templates (global, interface, connector, carrier and recipient)
Messages	Exports the messages table of the database.
Advanced ASCII Polling	Exports the Advanced Polling settings for the GetASCII module.
On-Call Schedule	Exports the existing schedule of messages to send to on-call groups.
Ad-Hoc Map	Exports the range of PIN numbers set for use by the ad-hoc function.
Mailboxes	Exports the mail settings set for each recipient or group.
Lookup Table	Exports the Lookup Table settings used by the GetASCII and GetSerial interfaces.
PinInfo	Exports a list of all recipient names and ID/PIN values.

This section allows you to import information in to PageGate's database.



Menu Item	Function
All	This will import all items from the specified folder.
Settings	Imports the global settings for PageGate, including registration and Program Settings .
Security	Imports the Security Settings and Users .
Connectors	Imports the Connector settings.
Carriers	Imports a list of Carriers and their respective settings.
Recipients	Imports a list of Recipients
Groups	Imports a list of groups and group members.
Display Groups	Imports a list of display groups and display group members.
Templates	Imports all templates from the specified folder.
Messages	Imports messages from the specified import file.

PageGate

Advanced ASCII Polling	Imports the Advanced Polling settings for the GetASCII module.
On-Call Schedule	Imports a schedule to send messages to on-call groups.
Ad-Hoc Map	Imports a range of IDs for use with PageGate's ad-hoc feature.
Mailboxes	Imports information about the mail settings for each recipient or group.
Lookup Table	Imports the Lookup Table settings used with the GetASCII and GetSerial interfaces.
Recipients (simplified)	A simplified form of the Recipients and Groups import file.

File Formats

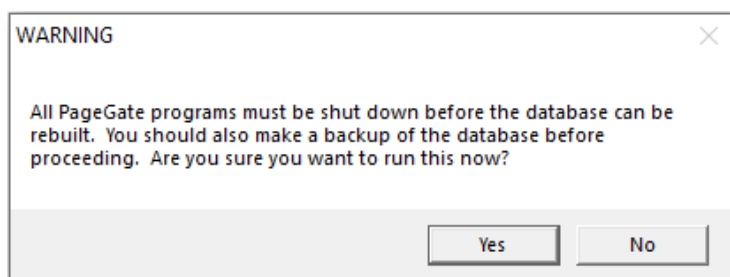
PageGate uses a standardized Comma Separated Values (CSV) file format for exporting and importing. If you want to import information into PageGate, we would recommend creating some sample or dummy data and then performing an export.

Each export file contains the specific structure and format required to import, including version information and field names. PageGate also uses the .csv extension on its' exported data to make opening it in programs like Excel much easier. Field labels are contained on line 2 of the file.

To import information into PageGate, please follow the guideline shown in the export files.

Rebuild Database

This option will compact and rebuild the PageGate database. When you click on the Rebuild Database option, you will receive the following prompt:

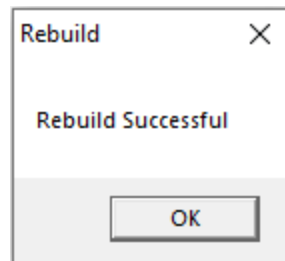


All of PageGate's modules **must** be stopped in order to rebuild the database. This can be accomplished by clicking on the pager icon in the Windows system tray (next to the clock) and choosing 'Quit' or 'Stop'.

PageGate Admin: How to Configure PageGate

No part of PageGate can be running when this rebuild takes place. Please be sure to shut down all instances of the GUI Client and PageGate Admin.

When you click on Yes, the rebuild will take place. After the rebuild has finished, you should receive this prompt:



After you click on OK, it is safe to run the PageGate Server, Admin and Client once more.

If PageGate is running as an Application, restart PageGate by choosing 'PageGate Server' from the Windows start menu.

If PageGate is running as a Windows Service, click on the pager icon in the Windows system tray (next to the clock) and choose 'Start' from the popup menu.

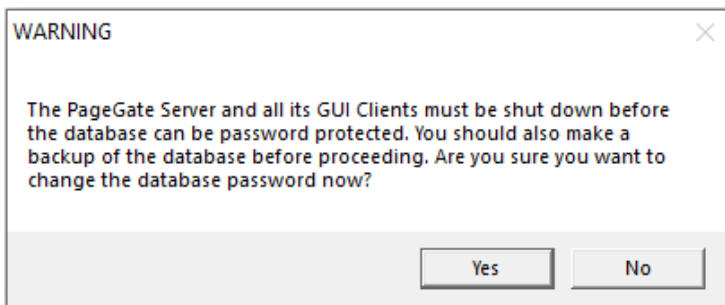
Note: An external rebuild utility is included with PageGate. It is called 'Jetcomp.exe' and can be found in the PageGate program directory. This utility can be used in cases where the PG Admin program is unable to start due to certain types of database corruption. After the database is rebuilt, you will have to restart the PageGate Server module.

Password Protect

The password protect option allows you to set a password specifically on the database. This does not have anything to do with the other [Security](#) features of PageGate; Entering a database password will restrict access to the database from programs like Microsoft Access.

PageGate Admin: How to Configure PageGate

When you click the Password Protect option, you will receive this prompt:

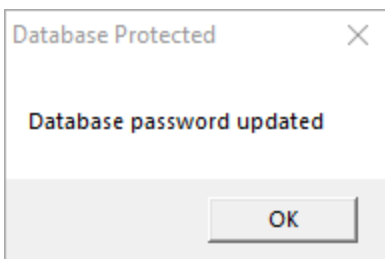


When you click on Yes, you will be prompted with this:



Enter a password, then click on Apply.

When you receive the following prompt:



Click OK for the password to be accepted.

NOTE: Without the password, the database cannot be opened. If this password is set, the next time each PageGate GUI Client is started, the you will need to enter it (you will be prompted for it automatically). The

PageGate Admin: How to Configure PageGate

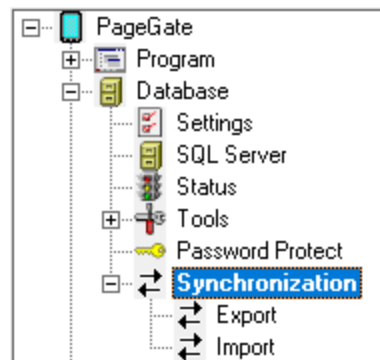
PageGate GUI Client will only prompt for this password once, and then it will be stored for the future (unless the password is changed again).

DO NOT LOSE THIS PASSWORD.

If this password is lost, the database may become permanently inaccessible!

Synchronization

This section allows you to configure multiple PageGate servers to synchronize their data with each other.



- [Export](#)

This section should be used on the primary PageGate server that is going to push updates out to the secondary PageGate server(s).

- [Import](#)

This section should be used on the secondary PageGate server(s) that should receive updates from the primary PageGate server.

Export

NOTE: This option should only be configured on the Master PageGate Server. Do NOT configure the Synchronization Export on Secondary PageGate Servers.

PageGate Admin: How to Configure PageGate

6

The screenshot shows a configuration panel with the following elements:

- Enable automatic export
- 60 Export interval (minutes)
- Export folder 1
- Export folder 2
- Export folder 3
- Export folder 4
- Export folder 5
- Export folder 6
- Export folder 7
- Export folder 8
- Export carriers
- Export recipients
- Export groups
- Export security
- Apply button

Option	Function
Enable automatic export	This option will export changes made to the master database to each of the slave databases.
Export interval (minutes)	This field sets the interval at which changes are exported to the slave databases. It is not recommended to have this set lower than once per hour.
Export folder 1 - 8	This field sets the path to the database directory of the secondary servers.
Export Carriers	This option will export the carriers from the master database to the secondary databases. This includes carrier templates.
Export Recipients	This option will export the recipients from the master database to the secondary databases. This includes recipient templates, mailboxes and ad-hoc settings.
Export Groups	This option will export groups from the master database to the secondary databases. This includes group members, mailboxes, ad-hoc settings and on-call schedules.
Export Security	This option will export the security settings and users from the master database to the secondary databases.

NOTE: This option should only ever be set on the Slave PageGate Servers. Do NOT configure the Synchronization Import on the Primary PageGate Server.

Option	Function
Enable automatic import	Enabling this option will import changes made to the master database into each the slave database this copy of the Admin controls.
Import interval (minutes)	This field sets the interval at which changes are imported to the slave database. It is not recommended to have this set lower than once per hour.
Import Carriers	Enabling this option will import the carriers, including their templates) from the master database.
Import Recipient	Enabling this option will import the recipients (including templates, mailboxes and ad-hoc settings) from the master database.
Import Groups	Enabling this option will import groups (including group members, mailboxes, ad-hoc settings and on-call schedules) from the master database.
Import Security	Enabling this option will import the security settings and users from the master database.

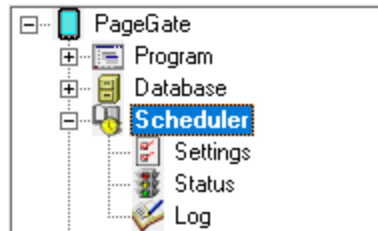
When enabling the option to import Carriers, Recipients, Groups, or Security Settings, you will be presented with four options:

PageGate Option

Option	Function
Add/Update/Remove	This option will allow the imported entries to Add new recipients, Updated existing recipients, or Remove existing recipients that are not on the list of recipients to be imported.
Add Only	This option will only allow new recipients to be added upon import.
Update Only	This option will only update existing recipients upon import.
Add/Update	This option will allow new recipients to be created and existing recipients can be updated upon import.

Scheduler

The PageGate Scheduler is the main, core processing component of the PageGate server. While there aren't many settings controlled by this section of the PG Admin, it is incredibly important to run the PageGate Scheduler as all functions for all APIs and Connectors are governed by the Scheduler.



- [Settings](#)

This section controls how often the PageGate Scheduler polls the PageGate database.

- [Status](#)

This section displays the running status of the PageGate Scheduler.

- [Log](#)

This section contains a log of everything recorded by the PageGate Scheduler.

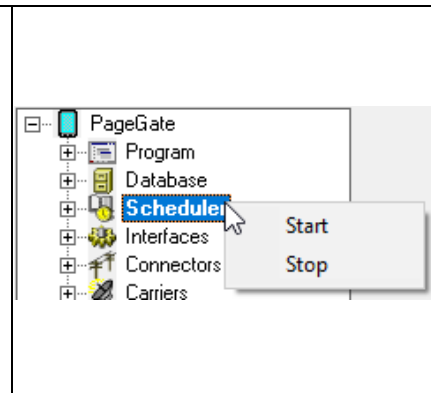
Right clicking on Scheduler will provide a context menu with two options.

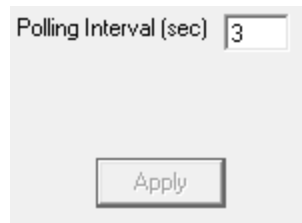
- [Start](#)

Selecting this option will start the PageGate Scheduler, if stopped.

- [Stop](#)

Selecting this option will stop the PageGate Scheduler, if started.





In the Settings of the PageGate Scheduler, you can modify the scheduler's polling interval. This value represents how often, in seconds, the scheduler should poll the database for new information but does not affect how often the scheduler updates the database when processing messages.

Status



The status subsection of Scheduler will show you if the PageGate Scheduler is currently running or

stopped. There are three ways to start the scheduler:

- 1) Restart the PageGate Server application or service.
- 2) Reboot the computer.
- 3) To manually stop or start the PageGate Scheduler, right click on the word Scheduler.

When you do, it will come up with the Start and Stop options.

Note: If the PageGate Scheduler is not enabled in the ["Run on this server"](#) section of the [PageGate Admin's program settings on the PageGate server](#), the scheduler will not be able to run and PageGate will not function properly.

Log

If logging is enabled, this log shows a summary of the events generated by the PageGate Scheduler.



Button	Function
Refresh	Displays the most current entries in the global log.
Print	Sends a copy of everything listed in the Scheduler Log to a printer.
Find	Allows you to search for a variable (such as a carrier or recipient name).
Purge	Deletes the entirety of the Scheduler Log.
Clipboard	Copies the contents of the Scheduler Log to the Windows Clipboard, allowing it to be pasted into another application.

Templates

Templates control quite a bit in PageGate. Some template variables control how messages are formatted while others determine how message negotiations should be encoded.

PageGate comes preloaded with a set of default templates that you may never need to modify. However, the Templates section provides administrators easy access to modify most parts of both the input to and output from PageGate.

PageGate Admin: How to Configure

PageGate

Message templates are available at the [Global](#), [Interface](#), [Connector](#), [Carrier](#) and [Recipient](#) levels. This means that a template can be configured for all messages that leave the system or just to a specific interface, connector, carrier or recipient.

The order of importance for the templates is as follows:

- 1) Recipient
- 2) Carrier
- 3) Connector
- 4) Interface
- 5) Global

This inter-connected system enables changes to be made for the maximum number of recipients by making a single change to one of the parent templates.

For example, if a Recipient has a template that has been applied to it, PageGate will format any messages delivered to this recipient based on its template regardless of whether Carrier, Connector, Interface or Global templates have been configured.

This allows you to create exceptions for Global rules by creating templates unique to a given Recipient, Carrier, Connector or Interface while the rest of your traffic remains formatted to a standard.

Both hard-coded text and PageGate specific macros can be used in each template, combining both static and dynamic elements into the message text.

Templates are comprised of three sections: [Settings](#), [Subject](#) and [Text](#).



The image shows a vertical stack of three form sections. The top section is labeled 'Settings' and contains a large empty text area with a vertical scrollbar on the right and horizontal scrollbars at the top and bottom. The middle section is labeled 'Subject' and contains a smaller empty text area with horizontal scrollbars at the top and bottom. The bottom section is labeled 'Text' and contains a large empty text area with a vertical scrollbar on the right and horizontal scrollbars at the top and bottom.

All Templates have three sections:

- [Settings](#)



The image shows a single form section labeled 'Settings'. It contains a large empty text area with a vertical scrollbar on the right and horizontal scrollbars at the top and bottom.

Settings variables (also called parameters) are used to turn settings on and off and are used to control the behavior of whatever you're interacting with.

The Settings template is unique in that variables specified in this section are cumulative. So, if you have Settings variables specified in the Global template and have completely different variables specified in a Carrier level template, all Settings specified are respected. However, if you have the same variable specified by multiple Settings templates, the highest order template's value will always be used.

For more information, see the [Template Settings section](#).

- [Subject](#)

PageGate Admin: How to Configure

PageGate



NOTE: Methods of delivery ([protocols](#)) that do not use subject lines do not reference this section of the template in any way.

The Subject section of a template controls the behavior of any subject line processed by this template. [Macros](#) can be used to insert information in to the subject line, like the date, time or other pertinent information. This template cannot use or reference parameters and will only accept [macros](#).

Subject templates will override templates of a lower order. For example, a Recipient level subject template will override any Carrier, Connector, Interface or Global subject.

For more information, see the [Subject Template section](#).

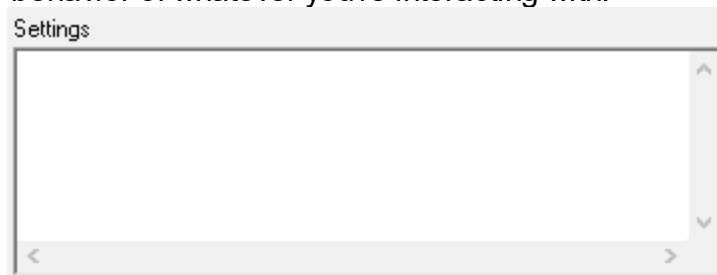
- [Text](#)



The Message section of the template controls the output and format of the body of any message controlled by this template and references Macros. A [Macro variable](#) is an inserted value to control, modify and specify the output of a message. The [Message](#) section of a Template should not use or reference any Setting variables and must only use [Macro variables](#). Text templates will override templates of a lower order. For example, a Recipient level text template will completely replace the subject line for that recipient and will override any Carrier, Connector, Interface or Program level Subject template you've applied.

Settings

Settings variables (also called parameters) are used to turn settings on and off and are used to control the behavior of whatever you're interacting with.



PageGate

For example, not every API uses the same language or data structure. So, when you're configuring

As an example, Airlink and Multitech both produce lines of cellular devices that can be accessed via COM or TCP but each implements a different AT command set. PageGate can automatically negotiate most connections with most devices and does automatically retain which command is best to use but you can decrease the overall amount of time it takes to deliver a message by specifying the necessary value in the Settings template.

Template values are ranked in this order:

- 1) Recipient
- 2) Carrier
- 3) Connector
- 4) Interface
- 5) Global

So, a Settings variable specified on a Carrier level will override any Global value. To use our earlier example, let's say you have an environment in which you're using multiple cellular modems; some of them Airlink, some of them Multitech. Since you have two devices that need to use the same protocol (GSM-AT-IP) but also need it implemented in different ways, you would specify a carrier level template value that configures the Airlink device for one AT command set and the Multitech device for the other. Alternately, you could tie each cellular gateway to a specific [Connector](#), then specify a [Connector level template](#) instead.

Variables that can be specified in the Settings section of a template can be broken down into a few categories: Protocol Delivery and Content Modification

Protocol Delivery

These variables modify the behavior of a protocol and specify certain declarations that must be made during the protocol negotiation sequence.

[Parameters for SMTP](#)

[Parameters for GSM-AT-IP](#)

PageGate Admin: How to Configure

PageGate

[Parameters for GSM-AT](#)

[Parameters for HTTPGET](#)

[Parameters for HTTPPOST](#)

[Parameters for FILE](#)

[Parameters for SNPP](#)

[Parameters for WCTP](#)

[Parameters for TAP](#)

[Parameters for RawTCP](#)

[Parameters for RawSerial](#)

[Parameters for XMPP](#)

[Parameters for TONE](#)

[Parameters for UCP](#)

Subject

The Subject section of a template controls the behavior of any Subject Line processed by this template.



Since Subject Templates may be applied on a Global, Carrier, Connector or Recipient level, it is entirely possible to create a standard format for all message subjects using a [Global Template](#), then specify exceptions to this global rule in the [Carrier](#), [Connector](#) or [Recipient](#) templates.

Generally speaking, the [SMTP or Email protocol](#) is the only protocol that actively references the Subject line during its protocol negotiation sequence. As such, you may want to format special Subject Templates for each of your SMTP carriers but that is not required. This can be also accomplished by a single [Global Template](#), if you want to apply these settings to all messages.

When configuring a Subject template, you can use plain text, macros or a mix and match of both. Please see the [Macros](#) section for a list of variables that can be used in this section.

For example, you could format a subject template like this:

Message from %Sender%

and any message processed by this template would include the text "Message from TheSenderValue" in the subject.

PageGate Admin: How to Configure PageGate

You could also use only plain text to configure a static subject. For example:
Message from CAD

Or you could use only Macros. For example:
%Date% %Time%

Text

The Message section of the template controls the output and format of the body of any message controlled by this template.



Since Message Templates may be applied on a Global, Carrier, Connector or Recipient level, it is entirely possible to create a standard format for all Messages using a [Global Template](#), then specify exceptions to this global rule in the [Carrier](#), [Connector](#) or [Recipient](#) templates.

When configuring a Text template, you can use plain text, macros or a mix and match of both. Please see the [Macros](#) section for a list of variables that can be used in this section.

For example, you could format a text template like this:
%Date% %Time% %Message%

and any message processed by this template would have the date and time stamp before the body of the message.

You could also completely override messages processed by this template by using plaintext and not referencing the %Message% variable. For example:
This is a message from your automated messaging system.

or

You have received a notification on %Date% at %Time%.

Macros

This is the master list of all Macro variables that can be used in a [Subject](#) or [Text](#) Template.

PageGate Admin: How to Configure PageGate

6

Variable	Description
%InternetDomain%	This variable inserts the setting specified Internet Domain Setting.
%Subject%	This variable inserts the Subject text.
%SubjectDelimiter%	This variable inserts the Subject Delimiter.
%Message%	This variable inserts the body of the Message.
%Sender%	This variable inserts the Sender of the Message.
%SenderMailbox%	If the Sender value is an email address, this variable inserts only the portion of the Sender's Address before the @.
%SenderDomain%	If the Sender value is an email address, this variable inserts only the portion of the Sender's Address after the @.
%Recipient%	This variable inserts the Recipient field of a Recipient's record or Group field of a Group record.
%RecipientFullName%	This variable inserts the Full Name field of a Recipient's record or the Description field of a Group record.
%OriginalRecipient%	This variable inserts the intended Original Recipient of a message. For example, if the message was originally addressed to a group, this will be the group's name.
%LastRecipient%	This variable inserts the Last Recipient a message delivery attempt was addressed to. this can be used when a message has "failed over" to another recipient, to insert the previous recipient's name.
%RecipientId%	This variable inserts the ID/PIN field of a Recipient's record.

PageGate

%Recipient Mailbox%	If the Recipient of a message is an email address, this variable inserts only the portion of the Recipient's Address before the @.
%Recipient Domain%	If the Recipient of a message is an email address, this variable inserts only the portion of the Recipient's Address after the @.
%Comment 1%	This variable inserts the Comment 1 field of a Recipient or Group's Record.
%Comment 2%	This variable inserts the Comment 2 field of a Recipient or Group's Record.
%Comment 3%	This variable inserts the Comment 3 field of a Recipient or Group's Record.
%Comment 4%	This variable inserts the Comment 4 field of a Recipient or Group's Record.
%Date%	This variable inserts the Date as specified by the operating system.
%Time%	This variable inserts the Time as specified by the operating system.
%MilitaryTime%	This variable inserts the Time as specified by the operating system output in a 00:00 - 24:00 format.
%Requested Date%	This variable inserts the Date on which a request to deliver a message was requested.
%Requested Time%	This variable inserts the Time on which a request to deliver a message was requested.
%Requested MilitaryTime%	This variable inserts the Time on which a request to deliver a message was requested output in a 00:00 - 24:00 format.

PageGate Admin: How to Configure PageGate

6

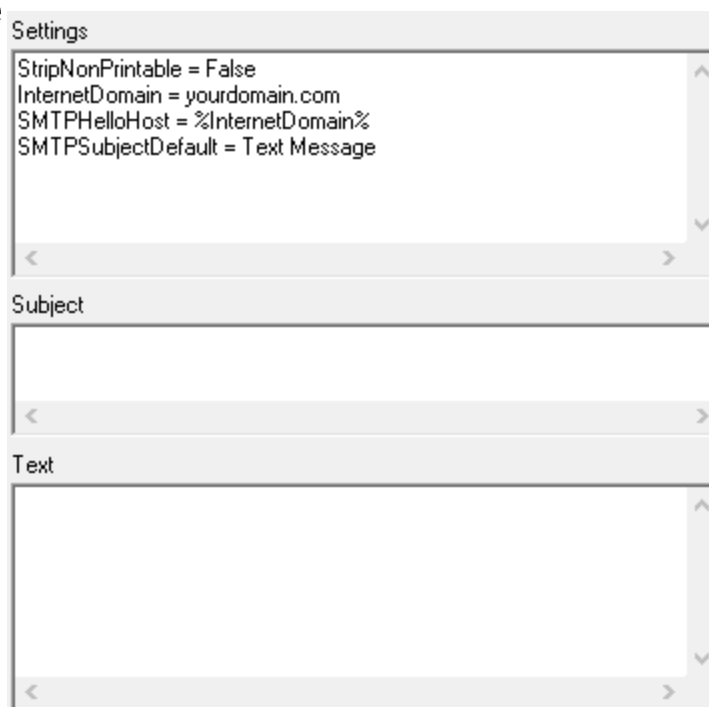
%TimeZone%	This variable inserts the Time Zone as specified by the operating system.
%ShortTimeZone%	This variable inserts the Time Zone as specified by the operating system output in short code.
%Station%	This variable inserts the PageGate Station ID value.
%LoginName%	This variable inserts the Windows Login Name of the user who performs a given action (such as sending a message).
%ComputerName%	This variable inserts the Windows Computer of the a computer that performs a given action (such as sending a message).
%Reliapage%	PageGate will sequentially number each message it sends to recipients. This is useful for detecting if a message has been missed (a number is missing).

Global Templates

The Global Template controls the behavior of all messages processed by the PageGate Server. PageGate's Global Template is contained in the Program - Template section of PG Admin.

PageGate Admin: How to Configure PageGate

6



When [configuring PageGate to send Email/SMTP](#), it is incredibly important to properly configure certain global variables. There are also some template variables that modify certain commands and responses to commands in certain protocol negotiations as well.

PageGate is pre-loaded with a default template that provides most base functionality, so you may not need to modify the Global Template. However, it is very, very important to make sure that this Template is formatted properly.

To format the Global Template, please review the [Template Components](#) section of the documentation.



The screenshot displays a configuration interface for carrier templates. It consists of three vertically stacked text input fields, each with a title and a scrollable area. The top field is titled 'Settings' and is empty. The middle field is titled 'Subject' and is also empty. The bottom field is titled 'Text' and is empty. Each field has a light gray border and a scroll bar on the right side.

Carrier level templates are most often used to apply specific settings or formatting requirements to your method of delivery and most of the variables you'll apply will be in the Settings section. For more information, please review the [Template Components](#) section of the documentation. Applying a custom template to a carrier tells PageGate to format messages delivered via this carrier in a certain way. Most carriers do not require a custom template but there may be things you want to customize about the look, output or specific format of your messages.

All delivery protocols have specific variables that can be used in the Settings section of a corresponding carrier template. These variables can be broken down in to a few categories: Protocol Delivery and Content Modification

Protocol Delivery

These parameters modify the behavior of a protocol and specify certain declarations that must be made during the protocol negotiation sequence.

[Parameters for SMTP](#)

[Parameters for GSM-AT-IP](#)

[Parameters for GSM-AT](#)

[Parameters for HTTPGET](#)

[Parameters for HTTPPOST](#)

[Parameters for FILE](#)

PageGate Admin: How to Configure

PageGate

[Parameters for SNPP](#)

[Parameters for WCTP](#)

[Parameters for TAP](#)

[Parameters for RawTCP](#)

[Parameters for RawSerial](#)

[Parameters for XMPP](#)

[Parameters for TONE](#)

[Parameters for UCP](#)

Parameters for SMTP/Email

These variables control how PageGate sends SMTP/Email messages. Variables specified on a carrier level will override variables specified on a global level.

[Click here](#) for more information on how PageGate sends Email.

Variable	Possible Values	Description
BodyMax Chars	Numeric	Example: BodyMaxChars = 200 This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.
Envelope From	Email Address	Example: EnvelopeFrom = your@sender.com Every email has two sender values, one on the 'outer' envelope and one on the 'inner' envelope. This setting will only change the outer envelope sender value. When this variable is not present, the outer and inner envelope sender value is the same. This variable provides a way to use a different outer envelope sender value, if required.
Envelope FromAsEmail	True False	Example: EnvelopeFromAsEmail = True This variable forces the EnvelopeFrom value to be turned into an valid email address, if it isn't already.
Filter	File path to and file name of the filter to run	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: Filter = c:\PageGateData\Filters\yourfilter.xml

		<p>It is entirely possible to call on PageGate's filtering system when delivering an SMTP/Email message, allowing you to reformat messages that enter the queue to any data structure you need. As an example, you could have a filter take every message and put it in an XML payload for delivery.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Email Address	<p>Example: From = your@sender.com</p> <p>This variable overrides any sender information passed in by the interfaces and APIs.</p> <p>By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
InterMessageDelay	Number of Milliseconds	<p>Example: InterMessageDelay = 1000</p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered. This is sometimes required to stay below messaging thresholds for SMTP servers.</p>
InternetDomain	Email domain	<p>Example: InternetDomain = alternate.domain.org</p> <p>This variable is used when carrier a should declare that messages are coming from a domain other than the one specified in the global template. Generally speaking, this variable should be set at a global level unless an exception is being made.</p>
ReplyTo	Email Address	<p>Example: ReplyTo = YourEmail@domain.com</p> <p>If the reply email address should be something other than the 'From' value, this variable allows you to specify the necessary address.</p>
SmtpConnectTimeout	Number of Seconds	<p>Example: SMTPConnectTimeout = 40</p> <p>This variable specifies the number of seconds PageGate should wait for a response from an SMTP host.</p>
SmtpForceAuth	login plain	<p>Example: SmtpForceAuth = plain</p> <p>This variable forces PageGate to only use the specified authentication method, rather than selecting from the list of</p>

PageGate

	cram- md5 ntlm	supported authentication styles reported by the mail server. PageGate currently supports using LOGIN, PLAIN, CRAM-MD5 and NTLM authentication.
Smt pForc eStartTls	True False	Example: Smt pForceStartTls = True This variable forces a carrier to use STARTTLS for its SMTP transactions, regardless of the value specified in the 'Port' field of the carrier settings.
Smt pFrie ndlyName s	True False	Example: Smt pFriendlyNames = True Some SMTP hosts use SMTP 'Friendly Names' as a part of anti-spam monitoring policy. Settings this value to true tells PageGate to include the SMTP 'Friendly Name' information in this carrier's transactions.
Smt pHell oHost	Email domain	Examples: Smt pHelloHost = %InternetDomain%, Smt pHelloHost = hardcodedvalue.org During the SMTP negotiation, this value is used during the EHLO/HELO exchange. Typically speaking, you will want this value to mirror the InternetDomain value and this should also be set at a global level. Implementing this at a carrier level is intended to create exceptions to the global rule, telling this carrier it should use a different value.
Smt pQpE ncode	True False	Example: SMT PQpEncode = False By default, PageGate uses Quoted Printable encoding for SMTP/Email. However, some systems can't interpret this format and require plaintext, so 'QP Encoding' needs to be turned off.
Smt pSen dReset	True False	Example: Smt pSendReset = False By default, PageGate issues the SMTP RSET command, which reset's the sessions values, between each message. This parameter provides the ability to turn that behavior off.
Smt pTLS Version	Text	Example: Smt pTLSVersion = TLS 1.3 or higher This parameter configures the version of TLS/SSL that should be used by this carrier. Valid values are: <ul style="list-style-type: none"> • default • TLS 1.3

		<ul style="list-style-type: none"> • TLS 1.2 • TLS 1.1 • TLS 1.0 • SSL 3.0 • TLS 1.3 or higher • TLS 1.2 or higher • TLS 1.1 or higher • TLS 1.0 or higher
SmtptUseDkim	True False	<p>Example: SmtptUseDkim = True</p> <p>If you have PageGate configured to deliver email as a mail server and have created a DKIM record, set this value to true to have PageGate include the DKIM signature in delivered traffic.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>
StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.</p>
SubjectDefault	Text	<p>Example: SubjectDefault = Dispatch Alert</p> <p>This parameter is used to set the subject line of all messages delivered through this carrier to a single, standard value.</p>
SubjectDelimiter	Text	<p>Example: SubjectDelimiter = \013\010 ~ :</p> <p>This parameter tells PageGate to look in the message text for subject line information. Multiple objects may be searched for and you'll want to separate each search parameter with a pipe. ASCII values may be scanned for but you must prefix any ASCII code with a backslash (\).</p>

PageGate Admin: How to Configure PageGate

6

		<p>In the above example, that would scan for a carriage return+line feed or a tilde (~) or a colon (:) within the specified SubjectSearchLength.</p> <p>Subjects parsed by this parameter are overridden by the SubjectDefault parameter.</p>
SubjectFilter	File path to and file name of the filter to be run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: SubjectFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter allows you to specify a filter that will only be run on the subject line of traffic delivered through this carrier.</p>
SubjectFilterDebugFile	File path to and file name of the log to be written	<p>Example: FilterDebugFile = c:\PageGateData\Filters\CarrierSubjectFilterLog.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
SubjectMaxChars	Numeric	<p>Example: SubjectMaxChars = 20</p> <p>This parameter determines the maximum number of characters the subject line may contain.</p>
SubjectSearchLength	Numeric	<p>Example: SubjectSearchLength = 10</p> <p>This parameter determines how many characters in to the message PageGate should look for the configured delimiting character(s).</p>
To	Email address	<p>Example: To = email@address.gov</p> <p>This parameter forces all traffic to go to a single email address, overriding any interface or recipient values for the delivery address.</p>
UseAlternateDNS	True False	<p>Example: UseAlternateDNS = True</p> <p>PageGate has a few methods of performing DNS lookups. If you are experiencing issues with standard DNS lookups, setting this parameter to true will tell the program to use an alternate method for performing them.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p>

		Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.
--	--	------------------------------------------------------------------------------------------------------------------------------

Parameters for GSM-AT-IP

These variables control how PageGate interacts with cellular routers and gateways accessed by TCP or SSH. It is recommended that you configure these variables on either the Carrier or Connector level.

[Click here](#) for more information on how PageGate uses cellular gateways and routers.

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when sending and receiving SMS through this carrier. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = ALERTS</p> <p>This parameter overrides any sender information passed in by the interfaces and APIs.</p> <p>By specifying this variable, you're telling PageGate to use a single, standardized sender's name for messages delivered through this carrier.</p>
GsmCmgldelay	Number of	<p>Example: GsmCmgldelay = 1000</p>

PageGate

	Milliseconds	This parameter increases the amount of time PageGate will wait on a response from cellular hardware when checking for messages.
GsmCmglString	Text	<p>Example: GsmCmglString = AT+CMGL=UREAD</p> <p>This parameter overrides the normal 'check for messages' command. PageGate will use whatever AT command is supplied in this statement instead.</p>
GsmSetTextModeOnce	True False	<p>Example: GsmSetTextModeOnce = True</p> <p>This parameter specifies that PageGate should only send the command to enter text delivery mode once until Connector is restarted.</p> <p>This parameter should not be used with cellular modems that require the "Enter Text Delivery Mode" command be issued in each session.</p>
GsmUseCmgf	True False	<p>Example: GsmUseCmgf = True</p> <p>This parameter specifies that the AT+CMGF=1 command should be used to enter text delivery mode.</p> <p>If your modem supports AT+CMGF=1, set this variable to true for the carrier or connector using the IP capable cellular modem, router or gateway.</p>
GsmUseCmgs	True False	<p>Example: GsmUseCmgs = True</p> <p>This variable specifies that the AT+CMGS command should be used to deliver text messages.</p>
GsmUseCpoint	True False	<p>Example: GsmUseCpoint = True</p> <p>This parameter tells PageGate to use the CradlePoint command set to interact with the device.</p>
GsmUseGT	True False	<p>Example: GsmUseGT = False</p> <p>This parameter tells PageGate to wait for a '>' character after using the CMGS command. By default, this is true but to disable it, set the variable above.</p>
GsmUseMmmg	True False	<p>Example: GsmUseMmmg = True</p> <p>This variable specifies that the AT+MMMGS command should be used to deliver text messages.</p>

GsmUseSmsm2m	True False	<p>Example: GsmUseSmsm2m = True</p> <p>This variable specifies that the AT*SMSM2M command should be used to deliver text messages.</p> <p>This is usually used with Sierra Wireless/Airlink hardware.</p>
GsmUseWscI	True False	<p>Example: GsmUseWscI = True</p> <p>This variable specifies that the AT+WSCL=1,2 command should be used to enter text delivery mode.</p> <p>If your modem uses AT+WSCL=1,2 to enter SMS delivery mode, set this variable to true.</p>
InterCommandDelay	Milliseconds	<p>Example: InterCommandDelay = 500</p> <p>This variable inserts a pause, in milliseconds, in between each command used in the protocol negotiation with the hardware. This is used to make sure PageGate doesn't send commands faster than the device's command shell can receive them and is not required for all devices.</p>
InterMessageDelay	Number of Milliseconds	<p>Example: InterMessageDelay = 1500</p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered.</p> <p>It is entirely possible for PageGate to overload the cellular radio in your device by sending information too quickly. By default, PageGate inserts a 1.5 second pause between messages to ensure this doesn't happen but this variable may need to be adjusted for specific hardware requirements.</p>
ModemHangupCommand	Command Set String	<p>Example: ModemHangupCommand = AT0</p> <p>If your cellular modem, router or gateway requires a special parameter for disconnect/hang-up, use this variable to specify the necessary command. Default if not set: ATH0</p>
ModemWakeUpCommand	Command Set String	<p>Example: ModemWakeUpCommand = \013</p> <p>If your cellular modem, router or gateway requires something other than AT<CR> to wake up, use this variable to specify the necessary command.</p>
ReplyFilter	File path to and file name	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\replyfilter.xml</p>

PageGate Admin: How to Configure PageGate

6

	of the filter to run	This parameter calls on PageGate's filtering system using the " Check for Messages " routine to retrieve SMS from the device. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging .
ReplyFilterDebugFile	File path to and file name of the log to write	Example: FilterDebugFile = c:\PageGateData\Filters\replyfilterlog1.log When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.
StripAlpha	True False	Example: StripAlpha = True Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'
StripCharacters	See description	Example: StripCharacters = \~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True False	Example: StripNonPrintable = True Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
TcpProtocol	SSH TCP	Example: TcpProtocol = SSH This parameter configures what protocol negotiation method should be used. By default, PageGate uses TCP but to connect with SSH instead, specify the above variable.
To	A Phone Number	Example: To = 7818290500 This parameter forces all traffic to go to the specified phone number, overriding any interface or recipient values.
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

PageGate Admin: How to Configure PageGate

6

UseTelnet	True	Example: UseTelnet = True
	False	This variable tells PageGate to negotiate telnet terminal capabilities upon connection

Parameters for GSM-AT

These variables control how PageGate interacts with modems. We recommend setting these parameters at either the Carrier or Connector level.

[Click here](#) for more information on how PageGate uses cellular modems.

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when sending SMS through this carrier. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = ALERTS</p> <p>This parameter overrides any sender information passed in by the interfaces and APIs.</p> <p>By specifying this variable, you're telling PageGate to use a single, standardized sender's name for messages delivered through this carrier.</p>

PageGate

GsmCmglDelay	Number of Milliseconds	<p>Example: GsmCmglDelay = 1000</p> <p>This parameter increases the amount of time PageGate will wait on a response from cellular hardware when checking for messages.</p>
GsmCmglString	Text	<p>Example: GsmCmglString = AT+CMGL=UREAD</p> <p>This parameter overrides the normal 'check for messages' command. PageGate will use whatever AT command is supplied in this statement instead.</p>
GsmSetTextModeOnce	True False	<p>Example: GsmSetTextModeOnce = True</p> <p>This parameter specifies that PageGate should only send the command to enter text delivery mode once until Connector is restarted.</p> <p>This parameter should not be used with cellular modems that require the "Enter Text Delivery Mode" command be issued in each session.</p>
GsmUseCmgf	True False	<p>Example: GsmUseCmgf = True</p> <p>This parameter specifies that the AT+CMGF=1 command should be used to enter text delivery mode.</p> <p>If your modem supports AT+CMGF=1, set this variable to true for the carrier or connector using the IP capable cellular modem, router or gateway.</p>
GsmUseCmgs	True False	<p>Example: GsmUseCmgs = True</p> <p>This variable specifies that the AT+CMGS command should be used to deliver text messages.</p>
GsmUseCpoint	True False	<p>Example: GsmUseCpoint = True</p> <p>This parameter tells PageGate to use the CradlePoint command set to interact with the device.</p>
GsmUseGT	True False	<p>Example: GsmUseGT = False</p> <p>This parameter tells PageGate to wait for a '>' character after using the CMGS command. By default, this is true but to disable it, set the variable above.</p>
GsmUseMmmg	True False	<p>Example: GsmUseMmmg = True</p>

		This variable specifies that the AT+MMGS command should be used to deliver text messages.
GsmUseSmsm2m	True False	<p>Example: GsmUseSmsm2m = True</p> <p>This variable specifies that the AT*SMSM2M command should be used to deliver text messages.</p> <p>This is usually used with Sierra Wireless/Airlink hardware.</p>
GsmUseWscL	True False	<p>Example: GsmUseWscL = True</p> <p>This variable specifies that the AT+WSCL=1,2 command should be used to enter text delivery mode.</p> <p>If your modem uses AT+WSCL=1,2 to enter SMS delivery mode, set this variable to true.</p>
InterCommandDelay	Milliseconds	<p>Example: InterCommandDelay = 500</p> <p>This variable inserts a pause, in milliseconds, in between each command used in the protocol negotiation with the hardware. This is used to make sure PageGate doesn't send commands faster than the device's command shell can receive them and is not required for all devices.</p>
InterMessageDelay	Number of Milliseconds	<p>Example: InterMessageDelay = 1500</p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered.</p> <p>It is entirely possible for PageGate to overload the cellular radio in your device by sending information too quickly. By default, PageGate inserts a 1.5 second pause between messages to ensure this doesn't happen but this variable may need to be adjusted for specific hardware requirements.</p>
ModemHangupCommand	Command Set String	<p>Example: ModemHangupCommand = AT0</p> <p>If your cellular modem, router or gateway requires a special parameter for disconnect/hang-up, use this variable to specify the necessary command. Default if not set: ATH0</p>
ModemWakeupCommand	Command Set String	<p>Example: ModemWakeupCommand = \013</p> <p>If your cellular modem, router or gateway requires something other than AT<CR> to wake up, use this variable to specify the necessary command.</p>

PageGate Admin: How to Configure PageGate

6

ReplyFilter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\replyfilter.xml</p> <p>This parameter calls on PageGate's filtering system using the "Check for Messages" routine to retrieve SMS from the device. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging.</p>
ReplyFilter DebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\replyfilterlog1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>
StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.</p>
To	A Phone Number	<p>Example: To = 7818290500</p> <p>This parameter forces all traffic to go to the specified phone number, overriding any interface or recipient values.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p>

PageGate Admin: How to Configure PageGate

6

		Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.
--	--	------------------------------------------------------------------------------------------------------------------------------

Parameters for HTTPGET

These variables control how PageGate uses the HTTP GET method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses HTTP GET.

Templates for HTTP GET carriers tell PageGate what variables and data to declare in the URL passed to the API.

All variables are declared in the 'Text' section of the template as follows:

|<name1>|<value1>|

|<name2>|<value2>|

etc.

Each value you specify will need to correspond to a variable that the receiving API will recognize. For example, let's say we wanted to send a message to an API that looked for to, from, subject and text as variables. We would need to format the text template as follows:

|to|%RecipientId%|

|from|%Sender%|

|subject|%Subject%|

|text|%Message%|

NOTE: The default parameter delimiter is a pipe (|) but this may be changed with the ParameterDelimiter setting below.

Variable	Possible Values	Description
BodyMaxChars	Numeric	Example: BodyMaxChars = 200 This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.
Filter	File path to and file	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml

PageGate Admin: How to Configure PageGate

6

	name of the filter to run	This parameter calls on PageGate's filtering system when sending and receiving SMS through this carrier. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging .
FilterDebug File	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This parameter overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
HttpErrorResponse	Command String See Description	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a GET requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an exact match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word failed: HttpErrorResponse = failed</p> <p>This would scan for the word failed in the <Reply> tag: HttpErrorResponse = <Reply>failed</Reply></p> <p>This would scan for the word reject or the word failed in the <Reply> tag: HttpErrorResponse = reject <Reply>failed</Reply></p>

		<p>This would scan for the word failed OR the word reject OR the phrase object not found: <code>HttpErrorResponse = failed reject object not found</code></p> <p>Any response from the web server that includes the search term(s) will be treated as a failure.</p>
HttpProtocol	<p>HTTP S HTTP</p>	<p>Example: <code>HttpProtocol = HTTPS</code></p> <p>This parameter forces the use of HTTPS regardless of the port used to connect.</p>
HttpResponseTimeout	<p>Millise conds</p>	<p>Example: <code>HttpResponseTimeout = 1000</code></p> <p>This variable forces PageGate to wait for the specified number of milliseconds (1000 = 1 second) for a response to the HTTP communication request.</p>
HttpSuccessResponse	<p>Com mand String See Descr iption</p>	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a GET requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word success: <code>HttpSuccessResponse = success</code></p> <p>This would scan for the word Acknowledged in the <Reply> tag: <code>HttpSuccessResponse = <Reply>Acknowledged</Reply></code></p> <p>This would scan for the word hijinks or the word Acknowledged in the <Reply> tag: <code>HttpSuccessResponse =hijinks <Reply>Acknowledged</Reply></code></p> <p>This would scan for the word success OR the word hijinks OR the word actualization: <code>HttpSuccessResponse = success hijinks actualization</code></p>

PageGate Admin: How to Configure PageGate

6

		Any response from the web server that includes the search term(s) will be treated as a success.
InterMessageDelay	Milliseconds	<p>Example: InterMessageDelay = 1000</p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered. It could be possible for PageGate to send a large number of data requests too quickly and this prevents that from happening.</p>
ParameterDelimiter	See Description	<p>Example: ParameterDelimiter = !</p> <p>Message templates for HTTP GET carriers usually involve parameter + variable pairs that are declared in this format: parameter variable </p> <p>ParameterDelimiter allows you to change the 'wrapping' character from a pipe () to something else, like an exclamation mark (!).</p>
ResponseIncludeRequest	True False	<p>Example: ResponseIncludeRequest = True</p> <p>Setting this parameter to true tells PageGate to include the original request sent to the API when recording responses.</p>
ResponsePath	File Path	<p>Example: ResponsePath = c:\PageGateData\APIResponses\</p> <p>This parameter allows you to specify the folder that responses should be written to. If API responses do not need to be tracked, this parameter is not required.</p>
ResponseRecipient	Recipient Group	<p>Example: ResponseRecipient = groupname</p> <p>When the HTTP GET or POST transaction completes, the specified recipient or group will receive any responses from the receiving API/web host. For more information, please see the Two Way Messaging section of the documentation.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>

PageGate Admin: How to Configure PageGate

6

StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.</p>
To	Text	<p>To = 7818290500</p> <p>This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Parameters for HTTPPOST

These variables control how PageGate uses the HTTP POST method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses HTTP POST.

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when sending and receiving SMS through this carrier. This allows for</p>

PageGate Admin: How to Configure PageGate

6

		the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging .
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This parameter overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
HttpErrorResponse	Command String See Description	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a POST requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word failed: HttpErrorResponse = failed</p> <p>This would scan for the word failed in the <Reply> tag: HttpErrorResponse = <Reply>failed</Reply></p> <p>This would scan for the word reject or the word failed in the <Reply> tag: HttpErrorResponse = reject <Reply>failed</Reply></p> <p>This would scan for the word failed OR the word reject OR the phrase object not found: HttpErrorResponse = failed reject object not found</p>

		Any response from the web server that includes the search term(s) will be treated as a failure.
HttpPostType	HeadBody UrlHeadBody	<p>HttpPostType = HeadBody configures an HTTP POST carrier to provide header information followed by structured data, usually JSON or XML, in the body.</p> <p>HttpPostType = UrlHeadBody configures an HTTP POST carrier to pass parameters in the URL, more like the POST method. This is not a typical use of HTTP POST but is used/required by some APIs.</p>
HttpProtocol	HTTP S HTTP	<p>Example: HttpProtocol = HTTPS</p> <p>This parameter forces the use of HTTPS or HTTP regardless of the port used to connect.</p>
HttpResponseTimeout	Millisecons	<p>Example: HttpResponseTimeout = 1000</p> <p>This variable forces PageGate to wait for the specified number of milliseconds (1000 = 1 second) for a response to the HTTP communication request.</p>
HttpSuccessResponse	Comm and String See Description	<p>NOTE: This parameter's routine only engages when receiving a 200 class response from the web server.</p> <p>This variable tells PageGate to scan for the specified keywords or keyphrases in responses to a POST requests made by this carrier. If those keywords or keyphrases are found, the message is treated as failed or rejected, depending on what you specify.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an exact match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word success: HttpSuccessResponse = success</p> <p>This would scan for the word Acknowledged in the <Reply> tag: HttpSuccessResponse = <Reply>Acknowledged</Reply></p> <p>This would scan for the word hijinks or the word Acknowledged in the <Reply> tag: HttpSuccessResponse =hijinks <Reply>Acknowledged</Reply></p>

PageGate Admin: How to Configure PageGate

6

		<p>This would scan for the word success OR the word hijinks OR the word actualization: <code>HttpSuccessResponse = success hijinks actualization</code></p> <p>Any response from the web server that includes the search term(s) will be treated as a success.</p>
InterMessageDelay	Milliseconds	<p>Example: <code>InterMessageDelay = 1000</code></p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered. It could be possible for PageGate to send a large number of data requests too quickly and this prevents that from happening.</p>
ParameterDelimiter	See Description	<p>Example: <code>ParameterDelimiter = !</code></p> <p>Message templates for HTTP POST carriers usually involve parameter + variable pairs that are declared in this format: <code> parameter variable </code></p> <p>ParameterDelimiter allows you to change the 'wrapping' character from a pipe () to something else, like an exclamation mark (!).</p>
ResponseIncludeRequest	True False	<p>Example: <code>ResponseIncludeRequest = True</code></p> <p>Setting this parameter to true tells PageGate to include the original request sent to the API when recording responses.</p>
ResponsePath	File Path	<p>Example: <code>ResponsePath = c:\PageGateData\APIResponses\</code></p> <p>This parameter allows you to specify the folder that responses should be written to. If API responses do not need to be tracked, this parameter is not required.</p>
ResponseRecipient	Recipient Group	<p>Example: <code>ResponseRecipient = groupname</code></p> <p>When the HTTP POST transaction completes, the specified recipient or group will receive any responses from the receiving API/web host. For more information, please see the Two Way Messaging section of the documentation.</p>
StripAlpha	True False	<p>Example: <code>StripAlpha = True</code></p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>

PageGate Admin: How to Configure PageGate

6

StripCharacters	See description	Example: StripCharacters = \~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True False	Example: StripNonPrintable = True Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
To	Text	Example: To = 7818290500 This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

This is how to use the 'Text' section of an HTTP POST carrier:

Each value you specify will need to correspond to a variable that the receiving API will recognize. For example, let's say we wanted to send a message using [Twilio's API](#). We know they're looking that looked for to, from and body as variables. We would need to format the text template as follows:

```
|to|%RecipientId%|  
|from|%Sender%|  
|body|%Message%|
```

NOTE: | is a pipe character and is found on most keyboards above the backslash character, \.

Some APIs, like [MessageBird](#), require an authentication variable in the body of the POST and those are declared like any other variable.

Example:

```
|Authorization|AccessKey PasteYourAccessKeyHere|
```

```
|recipients|%RecipientId%|
```

PageGate Admin: How to Configure

PageGate

|originator|%Sender%|

|body|%Message%|

Parameters for FILE

These variables control how PageGate uses the FILE method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate outputs ASCII files with the FILE protocol.

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system outputting files with this carrier. This is often an integral part of two-way messaging.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This parameter overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
InterMessageDelay	Milliseconds	<p>Example: InterMessageDelay = 1000</p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered. It could be possible for PageGate to</p>

		send a large number of data requests too quickly and this prevents that from happening.
StripAlpha	True False	Example: StripAlpha = True Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'
StripCharacters	See description	Example: StripCharacters = \~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True False	Example: StripNonPrintable = True Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
To	Text	Example: To = 7818290500 This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID.
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

Parameters for SNPP

These variables control how PageGate uses the SNPP method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses SNPP.

Variable	Possible Values	Description
----------	-----------------	-------------

PageGate

BodyMax Chars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when using this carrier.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
InterCommandDelay	Milliseconds	<p>Example: InterCommandDelay = 500</p> <p>This variable inserts a pause, in milliseconds, in between each command used in the protocol negotiation with the hardware. This is used to make sure PageGate doesn't send commands faster than the device's command shell can receive them and is not required for all devices.</p>
InterMessageDelay	Milliseconds	<p>Example: InterMessageDelay = 1000</p> <p>This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.</p>
SNPPMessageOnly	True False	<p>Example: SNPPMessageOnly = True</p> <p>This variable tells PageGate to use the MESS command, instead of the DATA command, during the SNPP delivery sequence.</p>

StripAlpha	True	Example: StripAlpha = True
	False	Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'
StripCharacters	See	Example: StripCharacters = \~
	description	This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True	Example: StripNonPrintable = True
	False	Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
To	Text	Example: To = 7818290500 This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.
UseFilterDll	True	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.
	False	Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

Parameters for WCTP

These variables control how PageGate uses the WCTP method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses WCTP.

Variable	Possible Values	Description and Possible Values
BodyMaxChars	Numeric	Example: BodyMaxChars = 200 This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.

PageGate Admin: How to Configure PageGate

6

EnvelopeFrom	Email Address	<p>Example: EnvelopeFrom = your@sender.com</p> <p>WCTP has two potential sender values, one on the 'outer' transaction and one on the 'inner' payload.</p> <p>This setting will only change the outer sender value. When this variable is not present, the outer and inner envelope sender value is the same. This variable provides a way to use a different outer envelope sender value, if required.</p>
EnvelopeFromAsEmail	True False	<p>Example: EnvelopeFromAsEmail = True</p> <p>This variable forces the EnvelopeFrom value to be turned into an valid email address, if it isn't already.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when using this carrier.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text Value	<p>Example: From = CAD</p> <p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
InterCommandDelay	Milliseconds	<p>Example: InterCommandDelay = 500</p> <p>This variable inserts a pause, in milliseconds, in between each command used in the protocol negotiation with the hardware. This is used to make sure PageGate doesn't send commands faster than the device's command shell can receive them and is not required for all devices.</p>

InternetDomain	Email domain	<p>Example: InternetDomain = exampledomain.com</p> <p>This variable determines the domain name PageGate will pass during the WCTP negotiation sequence.</p> <p>If this variable has already been implemented for SMTP/Email, it does not need to be specified again at a carrier level unless the receiving WCTP host requires you to specify a domain name other than your email domain.</p>
InterMessageDelay	Millisecons	<p>Example: InterMessageDelay = 1000</p> <p>This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>
StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.</p>
To	Recipient ID	<p>Example: To = 7818290500</p> <p>This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

PageGate Admin: How to Configure PageGate

6

WCTPBase64Encode	True False	<p>Example: WCTPBase64Encode = False</p> <p>This variable tells PageGate whether or not to use base64 encoding in WCTP transmissions. If this variable is not present, the value is assumed to be true and base64 is used.</p>
WCTPResponsesTo	Email Address Leave Empty	<p>Example: WCTPResponsesTo = email@domain.com</p> <p>This variable tells PageGate to include the sendResponsesToID field in the message envelope. A value of 'none' will instead send an allowResponse field with the value of 'false'.</p>

Parameters for TAP

These variables control how PageGate uses the TAP method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses TAP.

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when using this carrier.</p>
FilterDebugFile	File path to	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p>

	and file name of the log to write	When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.
From	Text	This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.
IgnoreCD	True	Example: IgnoreCD = True
	False	This variable tells PageGate to ignore the Carrier Detect signal when establishing a connection.
InterMessageDelay	Milliseconds	This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.
ModemHangupCommand	Command Set String	Example: ModemHangupCommand = AT0 If your cellular modem, router or gateway requires a special parameter for disconnect/hang-up, use this variable to specify the necessary command. Default if not set: ATH0
ModemWakeupCommand	Command Set String	Example: ModemWakeupCommand = \013 If your cellular modem, router or gateway requires something other than AT<CR> to wake up, use this variable to specify the necessary command.
StripAlpha	True	Example: StripAlpha = True
	False	Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'
StripCharacters	See description	Example: StripCharacters = \~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).

PageGate Admin: How to Configure PageGate

6

StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.</p>
To	Recipient ID	<p>Example: To = 7818290500</p> <p>This variable overrides the recipient ID/PIN. By specifying this variable, you're telling PageGate to send all traffic sent by this carrier should only go to the specified ID.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Parameters for RawTCP

These variables control how PageGate uses the RawTCP method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses RawTCP.

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when using this carrier.</p>

FilterDebug File	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
InterMessageDelay	Milliseconds	<p>This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.</p>
PauseChar	Special Character	<p>Example: PauseChar = ~</p> <p>This parameter tells PageGate to pause for the specified PauseTime before continuing on and should be set to a character that does not show up in your message text.</p> <p>As an example, you could use the above example to declare that a tilde (~) should create a pause. You could then format the message template of a RawTCP carrier as follows: ~myusername\013~~mypassword\013~%Message%\013</p> <p>That would tell this carrier that it should connect, wait 1 second, pass a username value and a carriage return, wait 2 seconds, pass the password value and a carriage return, wait 1 second, then pass the message text and a carriage return.</p>
PauseTime	Milliseconds	<p>Example: PauseTime = 1000</p> <p>This parameter tells PageGate how long to pause when the PauseChar is encountered.</p>
RawTcpErrorResponse	See Description	<p>This parameter scans through response packets for the specified keywords or keyphrases to determine if an error has occurred during communication.</p>

<p>PageGate</p>		<p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe. All values are considered to have wildcards.</p> <p>For example, this will scan for the word reject but will also match rejecting, rejected, etc: RawTcpErrorResponse = reject</p> <p>This would scan for the words fail* or reject*: RawTcpErrorResponse = fail reject</p> <p>This would scan for the word failed* OR the word reject* OR the phrase *object not found*: RawTcpErrorResponse = failed reject object not found</p> <p>Any response from the TCP host that includes the search term(s) will be treated as a failure.</p>
<p>RawTcpResponseTimeout</p>	<p>Milliseconds</p>	<p>Example: RawTcpResponseTimeout = 1000</p> <p>This variable forces PageGate to wait for the specified number of milliseconds (1000 = 1 second) for a response to the TCP communication request.</p>
<p>RawTcpSuccessResponse</p>	<p>See Description</p>	<p>This parameter scans through response packets for the specified keywords or keyphrases to determine if a communication attempt has been successfully completed.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word success: RawTcpSuccessResponse = success</p> <p>This would scan for the words Acknowledged or acknowledged RawTcpSuccessResponse = Acknowledged acknowledged</p> <p>This would scan for the word success OR the word hijinks OR the word actualization: RawTcpSuccessResponse = success hijinks actualization</p>

		Any response from the web server that includes the search term(s) will be treated as a success.
StripAlpha	True False	Example: StripAlpha = True Setting this value to true will remove all alphabetical characters from messages delivered through this carrier.
StripCharacters	See description	Example: StripCharacters = ~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True False	Example: StripNonPrintable = True Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
TcpProtocol	SSH TCP	Example: TcpProtocol = SSH This parameter configures what protocol negotiation method should be used. By default, PageGate uses TCP but to connect with SSH instead, specify the above variable.
To	Recipient ID	Example: To = 7818290500 This variable overrides the recipient ID/PIN. By specifying this variable, you're telling PageGate to send all traffic sent by this carrier should only go to the specified ID.
UseTelnet	True False	Example: UseTelnet = True This variable tells PageGate to negotiate telnet terminal capabilities upon connection
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

Parameters for RawSerial

These variables control how PageGate uses the RawSerial method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses RawSerial.

PageGate Admin: How to Configure PageGate

6

Variable	Possible Values	Description
BodyMaxChars	Numeric	<p>Example: BodyMaxChars = 200</p> <p>This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.</p>
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml</p> <p>This parameter calls on PageGate's filtering system when using this carrier.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
From	Text	<p>Example: From = CAD</p> <p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>

PageGate

InterMessageDelay	Millisecon ds	<p>This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.</p>
PauseChar	Special Character	<p>Example: PauseChar = ~</p> <p>This parameter tells PageGate to pause for the specified PauseTime before continuing on and should be set to a character that does not show up in your message text.</p> <p>As an example, you could use the above example to declare that a tilde (~) should create a pause. You could then format the message template of a RawSerial carrier as follows: <code>~myusername\013~~mypassword\013~%Message%\013</code></p> <p>That would tell this carrier that it should connect, wait 1 second, pass a username value and a carriage return, wait 2 seconds, pass the password value and a carriage return, wait 1 second, then pass the message text and a carriage return.</p>
PauseTime	Millisecon ds	<p>Example: PauseTime = 1000</p> <p>This parameter tells PageGate how long to pause when the PauseChar is encountered.</p>
RawSerialErrorResponse	Se e Description	<p>This parameter scans through response packets for the specified keywords or keyphrases to determine if an error has occurred during communication.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an <u>exact</u> match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe. All values are considered to have wildcards.</p> <p>For example, this will scan for the word reject but will also match rejecting, rejected, etc: <code>RawSerialErrorResponse = reject</code></p> <p>This would scan for the words fail* or reject*: <code>RawSerialErrorResponse = fail reject</code></p> <p>This would scan for the word failed* OR the word reject* OR the phrase *object not found*: <code>RawSerialErrorResponse = failed reject object not found</code></p>

PageGate Admin: How to Configure PageGate

6

		Any response from the TCP host that includes the search term(s) will be treated as a failure.
RawSerialResponseTimeout	Milliseconds	<p>Example: RawTcpResponseTimeout = 1000</p> <p>This variable forces PageGate to wait for the specified number of milliseconds (1000 = 1 second) for a response to the TCP communication request.</p>
RawSerialSuccessResponse	See Description	<p>This parameter scans through response packets for the specified keywords or keyphrases to determine if a communication attempt has been successfully completed.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an exact match to what you enter.</p> <p>Multiple search terms may be used and must be separated by a pipe.</p> <p>For example, this would only scan for the word success: RawSerialSuccessResponse = success</p> <p>This would scan for the words Acknowledged or acknowledged RawSerialSuccessResponse = Acknowledged acknowledged</p> <p>This would scan for the word success OR the word hijinks OR the word actualization: RawSerialSuccessResponse = success hijinks actualization</p> <p>Any response from the web server that includes the search term(s) will be treated as a success.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove all alphabetical characters from messages delivered through this carrier.</p>
StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p>

PageGate Admin: How to Configure PageGate

6

		Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
To	Recipient ID	Example: To = 7818290500 This variable overrides the recipient ID/PIN. By specifying this variable, you're telling PageGate to send all traffic sent by this carrier should only go to the specified ID.
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

Parameters for XMPP

These variables control how PageGate interacts with XMPP chat systems. We recommend setting these parameters at either the Carrier or Connector level.

[Click here](#) for more information on how PageGate uses XMPP chat systems.

Variable	Possible Values	Description
BodyMaxChars	Numeric	Example: BodyMaxChars = 200 This parameter limits the length of the %Message% macro and does not override the carrier "Max Chars" setting.
Filter	File path to and file name of the filter to run	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: Filter = c:\PageGateData\Filters\yourcarrierfilter.xml This parameter calls on PageGate's filtering system when sending messages through this carrier. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging .
FilterDebugFile	File path to and file name of the	Example: FilterDebugFile = c:\PageGateData\Filters\outboundfilter1.log When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.

PageGate Admin: How to Configure PageGate

6

	log to write	
From	Text	<p>Example: From = ALERTS</p> <p>This parameter overrides any sender information passed in by the interfaces and APIs.</p> <p>By specifying this variable, you're telling PageGate to use a single, standardized sender's name for messages delivered through this carrier.</p>
InterCommandDelay	Milliseconds	<p>Example: InterCommandDelay = 500</p> <p>This variable inserts a pause, in milliseconds, in between each command used in the protocol negotiation with the chat server. This is used to make sure PageGate doesn't send commands faster than the chat server's command shell can receive them and is not required in most cases.</p>
InterMessageDelay	Number of Milliseconds	<p>Example: InterMessageDelay = 1500</p> <p>This variable inserts a pause, in milliseconds, in between each message delivery attempt.</p>
ReplyFilter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\replyfilter.xml</p> <p>This parameter calls on PageGate's filtering system using the "Check for Messages" routine to retrieve chat messages from the XMPP server. This allows for the sorting and modification of both outbound messages and replies and is an integral part of two-way messaging.</p>
ReplyFilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\replyfilterlog1.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you</p>

		tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'
StripCharacters	See description	Example: StripCharacters = \~ This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).
StripNonPrintable	True False	Example: StripNonPrintable = True Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this carrier.
To	A Phone Number	Example: To = 7818290500 This parameter forces all traffic to go to the specified ID, overriding any interface or recipient values.
UseFilterDll	True False	Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter. Example: UseFilterDll = True Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.

Parameters for TONE

These variables control how PageGate uses the TONE method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses TONE.

Variable	Possible Values	Description
ModemWakeupCommand	Text	Example: ModemWakeupCommand = Custom Variable Here This variable is used if a custom initialization string is required to engage the serial line.

PageGate Admin: How to Configure PageGate

6

From	Text	<p>Example: From = CAD</p> <p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>
To	Text	<p>Example: To = 7818290500</p> <p>This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.</p>
IgnoreCD	True False	<p>Example: IgnoreCD = True</p> <p>This variable tells PageGate to ignore the Carrier Detect signal when connecting to a serial line.</p> <p>If this variable is not present, a False value is assumed.</p>
InterMessageDelay	Milliseconds	<p>Example: InterMessageDelay = 1000</p> <p>This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.</p>

Parameters for UCP

These variables control how PageGate uses the UCP method of communication. It is recommended to set these parameters at the Carrier level.

[Click here](#) for more information on how PageGate uses UCP.

Variable	Possible Values	Description
ModemWakeupCommand	Text	<p>Example: ModemWakeupCommand = Custom Variable Here</p> <p>This variable is used if a custom initialization string is required to engage the serial line.</p>
From	Text	<p>Example: From = CAD</p> <p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.</p>

To	Text	<p>Example: To = 7818290500</p> <p>This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.</p>
IgnoreCD	True False	<p>Example: IgnoreCD = True</p> <p>This variable tells PageGate to ignore the Carrier Detect signal when connecting to a serial line.</p> <p>If this variable is not present, a False value is assumed.</p>
InterMessageDelay	Milliseconds	<p>Example: InterMessageDelay = 1000</p> <p>This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.</p>

Templates for Faxing

When sending a fax from PageGate, you may want to modify the output so that the information maintains the look of a facsimile transmission rather than raw text output.

To apply a template to a Fax or FaxSrv carrier, all you have to do is go in to the Template sub-section of that carrier. In the "Message" section, you can enter the template and all important variables.

The following is an **example** of a Faxing Template:

FROM: <Your Name here>

TO: %Recipient %
 %Comment 1%
 %Comment 2%
 %Comment 3%
 %Comment 4%

SENT: %Date% %Time%

SUBJECT: %Subject %

MESSAGE:

%Message%

After entering the template information, click on Apply.
 Interface Templates

Templates applied to a specific interface format, or re-format, all messages processed by this interface. Interface level variables will override the same variables specified on a

PageGate Admin: How to Configure

PageGate

Global level but will be overridden by variables specified at a Connector, Carrier or Recipient level.

The image shows a configuration window with three sections, each containing an empty text input field with scrollbars:

- Settings**: An empty text area at the top.
- Subject**: A single-line text input field in the middle.
- Text**: A larger multi-line text area at the bottom.

For example, if the Interface Template formats the message to standardize the Sender's Name to a static value, but a Connector, Carrier or Recipient template specifies a different value, messages passed through those Carriers or to those Recipients will use the Sender's Name variable from their own templates and ALL other variables from higher level templates.

- [Parameters for GetAscii](#)
- [Parameters for GetMail](#)
- [Parameters for GetSerial](#)
- [Parameters for GetTAP](#)
- [Parameters for GetWeb](#)

Parameters for GetAscii

Variable	Description
----------	-------------

InterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This applies a filter to processed data. This allows the interfaces to apply the initial layer of data sorting and collection and then applies the filter to the result.</p> <ul style="list-style-type: none"> • Standard polling: Filter is applied against temp file with recipient, sender and message • Advanced polling: Filter is applied against temp file with recipient, sender and message, and after lookup substitution
InterfaceFilterDebugFile	<p>Example: InterfaceFilterDebugFile = c:\PageGateData\Filters\GetAsciiFilter.log</p> <p>When using an interface filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
RawInterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: RawInterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This applies a filter to raw, unprocessed data. This applies the filter before the interface has a chance to read the data.</p> <ul style="list-style-type: none"> • Standard polling: Filter is applied against the raw text file before GetAscii tries to process it • Advanced polling: Filter is applied against each record that is found, before it is processed any further (record pattern matching done first)
RawInterfaceFilterDebugFile	<p>Example: RawInterfaceFilterDebugFile = c:\PageGateData\Filters\GetAsciiFilter.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the declared location and file name.</p>
UseFilterDll	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

PageGate Admin: How to Configure

6

PageGate Parameters for GetMail

These variables control some of the PageGate mail server's basic functions.

Variable	Description
AltParser	<p>Example: AltParser = True</p> <p>If this value is set to 'True', GetMail to examine the attachments of messages it receives IF the body of the message is completely blank. If the attachment is in an ASCII format (HTML, XML, etc), the program will use the contents of the file as the body of the message.</p>
AltParserStripTags	<p>Example: AltParserStripTags = True</p> <p>If this value is set to 'True', GetMail will remove formatting tags from the data parsed from attached files.</p> <p>This entry's function is contingent on the GetMail_Alt_Parser registry entry existing and being set to 'True'.</p>
ForceAdhocEmail	<p>Example: ForceAdhocEmail = True</p> <p>Enabling this parameter tells GetMail that mailbox value for every incoming message should be treated as ad-hoc. When this is enabled, PageGate will not try to match the incoming mailbox name with a recipient or group.</p>
InterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter after messages have been extracted from the incoming email and decoding has taken place/</p>
InterfaceFilterDebugFile	<p>Example: InterfaceFilterDebugFile = c:\PageGateData\Filters\GetMailFilter.log</p> <p>When using an interface filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
RawInterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: RawInterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter before messages have been extracted from the incoming email and decoding has taken place/</p>

PageGate Admin: How to Configure PageGate

6

RawInterfaceFilterDebugFile	<p>Example: RawInterfaceFilterDebugFile = c:\PageGateData\Filters\GetMailFilter.log</p> <p>When using a raw interface filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
SubjectFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: SubjectFilter = c:\PageGateData\Filters\YourSubjectFilter.xml</p> <p>This parameter calls on a filter that will only work with the contents of the subject line and will not influence the recipient(s) or body of the message.</p>
SubjectFilterDebugFile	<p>Example: SubjectFilterDebugFile = c:\PageGateData\Filters\GetMailSubjectFilter.log</p> <p>When using a subject filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
UseFilterDll	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Parameters for GetSerial

Variable	Description
InterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter after GetSerial has received and processed a message trigger, including any lookup substitutions.</p>
InterfaceFilterDebugFile	<p>Example: InterfaceFilterDebugFile = c:\PageGateData\Filters\GetSerialFilter.log</p> <p>When using an interface filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>

PageGate Admin: How to Configure PageGate

6

RawInterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter before GetSerial has received and processed a message trigger, including any lookup substitutions.</p>
RawInterfaceFilterDebugFile	<p>Example: RawInterfaceFilterDebugFile = c:\PageGateData\Filters\GetSerialFilter.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the declared location and file name.</p>
UseFilterDll	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Parameters for GetTAP

Variable	Description
InterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter after GetTAP has received and processed a message trigger, including any lookup substitutions.</p>
InterfaceFilterDebugFile	<p>Example: InterfaceFilterDebugFile = c:\PageGateData\Filters\GetTAPFilter.log</p> <p>When using an interface filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
RawInterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter before GetTAP has received and processed a message trigger, including any lookup substitutions.</p>

RawInterfaceFilterDebugFile	<p>Example: RawInterfaceFilterDebugFile = c:\PageGateData\Filters\GetTAPFilter.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the declared location and file name.</p>
UseFilterDll	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Parameters for GetWeb

Variable	Description
GroupLabelAfter	<p>Example: GroupLabelAfter = True</p> <p>When using the GUI List web paging template, setting this parameter to true tells the website that the (G) that designates a group should appear after the group name instead of before.</p>
InterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter after a web trigger has been received by GetWeb and after its' contents have been decoded.</p>
InterfaceFilterDebugFile	<p>Example: InterfaceFilterDebugFile = c:\PageGateData\Filters\GetWebFilter.log</p> <p>When using an interface filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
RawInterfaceFilter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: RawInterfaceFilter = c:\PageGateData\Filters\yourfilter.xml</p> <p>This parameter calls on the filter before a web trigger has been received by GetWeb and after its' contents have been decoded.</p>

PageGate Admin: How to Configure PageGate

6

RawInterfaceFilterDebugFile	<p>Example: RawInterfaceFilterDebugFile = c:\PageGateData\Filters\GetWebFilter.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the declared location and file name.</p>
UseFilterDll	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Connector Templates

Templates applied to a specific connector format, or re-format, all messages passed through this connector. Connector level variables will override the same variables specified on a Global or Interface level but will be overridden by variables specified at a Carrier or Recipient level.

Typically speaking, you'll set most parameters at the carrier and interface levels but when PageGate is interacting with multiple endpoints simultaneously, Connector templates can make logically organizing things a bit easier.

The image shows a screenshot of a settings window with three text input fields. The first field is labeled 'Settings', the second is labeled 'Subject', and the third is labeled 'Text'. Each field has a scroll bar on the right side and navigation arrows at the bottom.

For use with Cellular Modems and Gateways

Variable	Possible Values	Description
Filter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\YourConnectorFilter.xml</p> <p>This parameter calls on PageGate's filtering system when sending messages through this connector.</p>
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\connector1filter.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
InterMessageDelay	Number of Milliseconds	<p>Example: InterMessageDelay = 1500</p> <p>This variable inserts a pause, in milliseconds, in between each message that is delivered by this connector.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered through this carrier. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>
StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>

PageGate Admin: How to Configure PageGate

6

StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this connector.</p>
SubjectFilter	File path to and file name of the filter to run	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: SubjectFilter = c:\PageGateData\Filters\YourSubjectFilter.xml</p> <p>This parameter calls on a filter that will only work with the contents of the subject line and will not influence the recipient(s) or body of the message.</p>
SubjectFilterDebugFile	File path to and file name of the log to write	<p>Example: SubjectFilterDebugFile = c:\PageGateData\Filters\ConnectorSubject.log</p> <p>When using a subject filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
UseCarrierAuth	True False	<p>Example: UseCarrierAuth = False</p> <p>This setting tells the connector whether or not it should use authentication when connecting to the cellular modem, router or gateway. This is enabled by default.</p> <p>This is most often used when multiple makes and models of cellular modem, router or gateway are being used and some devices require authentication but others don't.</p> <p>This allows you to have a single carrier using the GSM-AT or GSM-AT-IP protocols and specialize each connector for the different types of devices in your environment.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p>

PageGate Admin: How to Configure PageGate

6

	<p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Recipient Templates

Templates applied to a specific Recipient format, or re-format, all messages passed to this Recipient. Recipient level variables will override the same variables specified in any other Template.

Recipient level parameters will not, and can not, be overridden by Global, Interface, Connector or Carrier templates.

The screenshot shows a configuration window with three text input fields. The top field is labeled 'Settings', the middle one 'Subject', and the bottom one 'Text'. Each field is currently empty and has a scroll bar on its right side.

For example, if you have From = CAD in Recipient Template, CAD is the only sender's name that will be used when messaging this recipient, regardless of what values the interface or carrier might try to use.

Variable	Possible Values	Description
Filter	File path to and file	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: Filter = c:\PageGateData\Filters\YourRecipientFilter.xml</p>

PageGate Admin: How to Configure PageGate

6

	name of the filter to run	This parameter calls on PageGate's filtering system when sending messages to this recipient.
FilterDebugFile	File path to and file name of the log to write	<p>Example: FilterDebugFile = c:\PageGateData\Filters\RecipientFilter.log</p> <p>When using a filter, you can use this variable to output this specific filter's log to the specific location and file name.</p>
StripAlpha	True False	<p>Example: StripAlpha = True</p> <p>Setting this value to true will remove any alphabetical character from messages delivered to this recipient. For example, if you tried to send the message 'Alert on node 127.0.0.1', the only thing delivered would be '127.0.0.1'</p>
StripCharacters	See description	<p>Example: StripCharacters = \~</p> <p>This parameter will remove all instance of any character(s) entered. In the above example, that would strip out all pipes (), backslashes (\) and tildes (~).</p>
StripNonPrintable	True False	<p>Example: StripNonPrintable = True</p> <p>Settings this value to true will remove any non-printable ASCII characters, like carriage returns, line feeds, horizontal tabs and similar, from messages delivered through this connector.</p>
SubjectFilter	File path to and file name of the filter	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: SubjectFilter = c:\PageGateData\Filters\YourSubjectFilter.xml</p> <p>This parameter calls on a filter that will only work with the contents of the subject line and will not influence the recipient(s) or body of the message.</p>

	to run	
SubjectFilterDebugFile	File path to and file name of the log to write	<p>Example: SubjectFilterDebugFile = c:\PageGateData\Filters\ConnectorSubject.log</p> <p>When using a subject filter, you can use this parameter to output this specific filter's log to the declared location and file name.</p>
UseCarrierAuth	True False	<p>Example: UseCarrierAuth = False</p> <p>This setting tells the connector whether or not it should use authentication when connecting to the cellular modem, router or gateway. This is enabled by default.</p> <p>This is most often used when multiple makes and models of cellular modem, router or gateway are being used and some devices require authentication but others don't.</p> <p>This allows you to have a single carrier using the GSM-AT or GSM-AT-IP protocols and specialize each connector for the different types of devices in your environment.</p>
UseFilterDll	True False	<p>Note: pgfilter.exe and pgfilter.dll must reside in the PageGate program directory to use this parameter.</p> <p>Example: UseFilterDll = True</p> <p>Setting this parameter to true tells PageGate to use the filter dll, which is a more efficient method of running the filter.</p>

Success and Failure Notification Templates

Note: This feature requires the GetMail Interface to be a part of your license.

This type of template is completely unlike the other Templates. These templates trigger a specific behavior within the program and are not contained within the Administrative GUI.

PageGate Admin: How to Configure

PageGate

This feature will allow you to configure PageGate to send notifications when a message succeeds or when a message fails.

When a message is successfully delivered, PageGate looks for a file named `successnotifylist.txt` in the PageGate program folder. This file can contain a list of email addresses that will receive notification that the message was successful. You can use the special value `%Sender%` on one of the lines to have the sender of the message in the list.

In addition you will need to create a file named `success.mtp` and place it in the PageGate program directory. This file is used as a template for what is sent in the success notification message. It uses the same variables as the [Macros](#).

The same process happens for a failed message, but it uses a `failurenotifylist.txt` and a `failure.mtp` file instead.

Create a `successnotifylist.txt` file in the PageGate folder with a single line containing `%Sender%`

Do the same for a `failurenotifylist.txt` file.

The `success.mtp` file should look like this:

Subject: Successful Message To %Recipient%

Message Info:

Recv: %RequestedDate% %RequestedTime%

Sent: %Date% %Time%

To: %Recipient%

From: %Sender%

Msg: %Message%

The `failure.mtp` file should look like this:

Subject: Failed Message To %Recipient%

Message Info:

Recv: %RequestedDate% %RequestedTime%

Fail: %Date% %Time%

To: %Recipient%

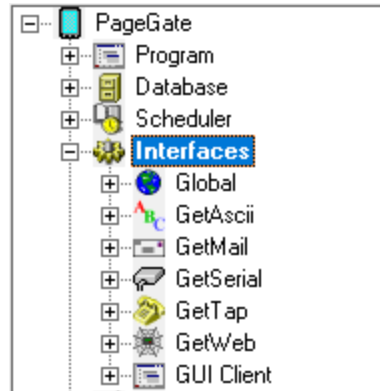
From: %Sender%

Msg: %Message%

PageGate Admin: How to Configure

PageGate Interfaces

The Interfaces section is where you'll configure your APIs to accept information from different sources. Each interface controls behavior for that API, so it's important to understand which API you want to implement. [Click here](#) for more information on how PageGate accepts messages.



The APIs that can be configured in the PageGate Admin are:

- [GetASCII](#)

Also known as the Command Line/ASCII interface, this interface is capable of monitoring a directory or series of directories on the local hard drive or across the network for ascii file output and also provides a command line interface for other applications to use. This module allows many 'off the shelf' and custom applications, such as SolarWinds' Network Performance Monitor, SolarWinds' Orion, Paessler's PRTG and several CAD programs, among many others, to tell PageGate when a message needs to be sent.

- [GetMail](#)

This interface allows you to receive SMTP/Email traffic and translate it in to outbound messages in a different format. For example, you could email a message to PageGate to deliver as a message to pagers, cell phones and/or other email addresses.

- [GetSerial](#)

This interface allows PageGate to monitor a serial/RS232 port, virtual or physical, for raw data text strings to be parsed in to outbound messages.

- [GetTap](#)

This interface gives PageGate the ability to monitor a serial port or modem to receive TAP transmitted pages, effectively allowing PageGate to function as a TAP receiver or a TAP terminal for any devices and systems that can transmit that protocol, such as a Motorola or Zetron paging terminal.

- [GetWeb](#)

This interface provides a CGI executable that can be hosted on a web server and a series of pre-formatted templates to publish a website messaging interface. Web Developers

PageGate Admin: How to Configure

PageGate

can use this web server interface to send messages to PageGate from their webpages (HTML, ASP, JavaScript, Java, etc.). This means that you can host messaging websites on your local intranet or, if GetWeb is interfaced with an outward facing web server, on the internet.

- [GUI Client](#)

A Windows based program installed on workstations on the same network as the PageGate server. This program provides a graphical user interface to send and schedule messages.

- [Additional APIs](#)

PageGate also supports additional APIs that are not controlled by the PageGate Admin. These include the [SNPP Server](#), [TAP IP Server](#), [TAP to ASCII](#) and [Serial to ASCII](#) modules.

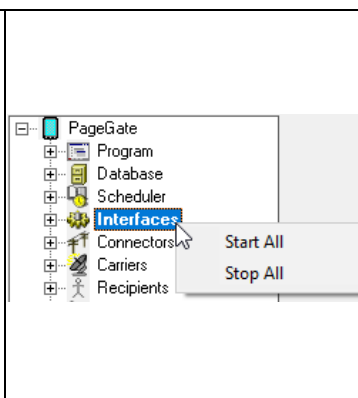
Right clicking on Interfaces will provide a context menu with two options.

- [Start All](#)

Selecting this option will start all configured and enabled Interfaces, if stopped.

- [Stop All](#)

Selecting this option will stop the all configured and enabled Interfaces, if started.



GetAscii

ASCII is, literally, text. What you're reading on the screen right now are ASCII characters. HTML, XML and JSON are examples of command formatted ascii text, which means that GetASCII can be configured to read in and recognize values in those formats as well. However, this API is one of PageGate's powerhouse interfaces and is used in a variety of implementations.

For example, if you use the SNPP Server or TAP over IP Server, these reference the GetASCII module. This module also powers the [Filter Pack](#) and can be implemented with the [2-way](#) reply system for [use with cellular infrastructure](#) to allow you much greater control over who receives what replies.

On the surface, GetASCII is incredibly easy to implement, especially if you're just having it read input from network monitoring programs and CAD systems. However, you can implement incredibly intricate and complex operations to process data from numerous locations, then format and reformat that data into any configuration want before delivering it.

PageGate Admin: How to Configure PageGate

For example, if you have a mapping data program that outputs an ascii log file, you can configure GetASCII to monitor that log for any mapping data, then feed that data through a [filter pack](#) to turn it in to an alert to your personnel.

It's also important to note that PageGate can use the [File](#) protocol to write files that GetASCII can process. This allows you to take information from other APIs, whether PageGate's or a web API, and pass it to the GetASCII module for message and data parsing.



- [Settings](#)

This section controls the implementation of the GetASCII module. This is where you configure all directories and text processing parameters.

- [Status](#)

This section displays the running/not running status of the interface.

- [Template](#)

This section allows you to configure a custom template for messages processed by this API.

- [Log](#)

This section is a record of all activity performed by the interface.

This interface allows PageGate to monitor a directory or series of directories on the local hard drive or across the network for ascii file. This API also provides a command line interface for other applications to use. This module allows many 'off the shelf' and custom applications, such as SolarWinds' Network Performance Monitor, SolarWinds' Orion, Paessler's PRTG and several CAD programs, among many others, to tell PageGate when a message needs to be sent.

In addition to the interface presented in the PageGate Admin, the GetASCII module provides two [command line interface](#) executables, allowing other programs to pass parameters to the GetASCII module directly. For more information, please see the [Command Line Interface](#) section of the documentation.

There are two ways to implement GetASCII: Standard Polling and Advanced Polling

PageGate

- [Standard Polling](#)

GetASCII has a standard file format and file naming convention that it will recognize without the need to configure specific parsing parameters. You'll need to configure the [basic polling directory](#) to tell PageGate where to look for these files. Once you have that specified, you can output a file with a .asc extension in the folder and PageGate will read it in, looking for the standard format. The file should be formatted as follows:

```
Recipient Value  
Sender Value  
Message Value
```

So, for example, we could create a file named test.asc, format it as shown below and drop it in the basic polling directory:

```
ITDept  
CAD  
This is a test
```

and that would send a message to the recipient or group named ITDept from the sender CAD and the message would say "This is a test".

Notes:

The basic polling directory must be configured to use Advanced Polling. Typically speaking, we recommend creating c:\PageGateData\ASCII\ as the basic polling folder.

It is not recommended to have the Standard and Advanced polling locations pointed at the same folder. This can cause file read errors if an advanced routine polls for files with the extension .asc

- [Advanced Polling](#)

When your data won't follow the standard file format or naming convention, you'll need to configure the Advanced Polling options to tell GetASCII how to interpret the ascii data it needs to handle. The Advanced Polling is used when you want to process free floating, HTML, XML, JSON or other ascii formatted data.

For more information, please see the [Advanced Polling](#) section of the documentation.

PageGate Admin: How to Configure PageGate

6

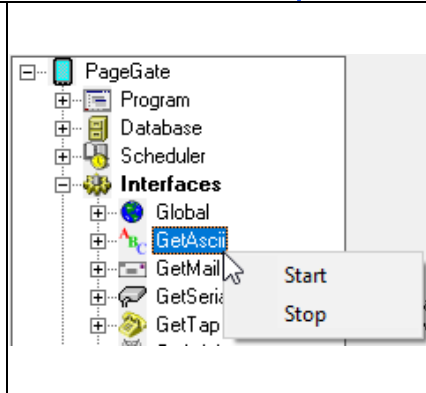
Right-clicking on GetASCII will provide a context menu with two options.

- Start

Selecting this option will start the GetASCII interface, if stopped.

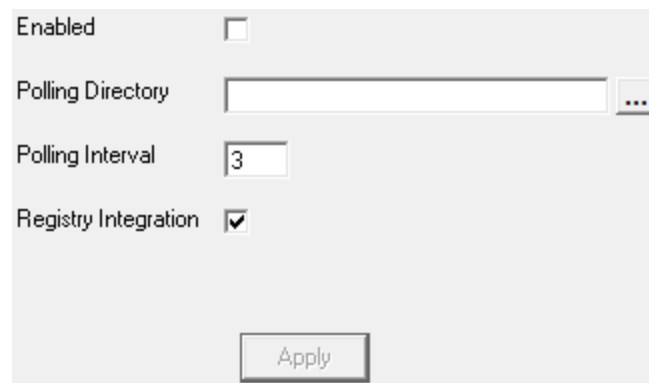
- Stop

Selecting this option will stop the GetASCII interface, if started.



Settings

This section and its sub-sections contain all configuration settings for the implementation of the GetASCII API.



The Settings section determines the basic polling directory. When using [Standard Polling](#), this is the directory that will host the [command line executables](#) as well as the directory to which you should output standard formatted files for processing.

PageGate

Setting	Function
Enabled	When the Enabled checkbox is checked, it tells the PageGate server to run the GetASCII module.
Polling Directory	This field specifies basic polling location to monitor for ascii files named something.asc to be processed. This value can be set to a local path (c:\PageGateData\ascii\, for example) or a UNC path (\\server\path\). If you're running PageGate as a Windows Service and GetASCII is referencing a UNC path, you must provide a service account to the PageGate GetASCII Windows service.
Polling Interval	This field determines how often the GetASCII module looks for new ascii files in the specified Polling Directory.
Registry Integration	When enabled, this tells GetAscii to write information to HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Note Page\PageGate for other applications and platforms to reference.

Expanding the Settings menu will display the following menu items:

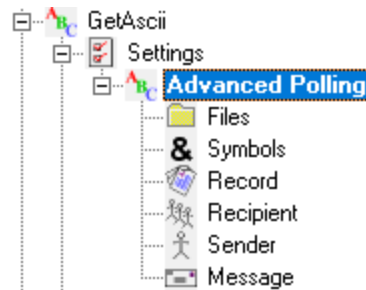


[Click here](#) to go to the [Advanced Polling](#) section of the documentation for more information.

Advanced Polling

This section is where you'll configure PageGate's advanced ascii parsing parameters. This is most often used when you need to process ascii data that doesn't follow [PageGate's standard format or file naming convention](#). This allows you to process XML, HTML, JSON and similarly styled content as well as the ability to process free floating data within an ascii file.

PageGate Admin: How to Configure PageGate



- [Files](#)

This section determines the directory or series of directories that GetASCII should look for content within and also specifies style of implementation as well as the naming convention of the files being worked with.

- [Symbols](#)

This section determines certain special characters and their functions.

- [Record](#)

This section is where you configure the pattern recognition to determine what represents the start and end of valid data to process.

- [Recipient](#)

This section is where you configure the pattern recognition to determine what represents who the messages should go to. This section can also be set to a static recipient, to indicate that all traffic processed from the serial line should be sent to the specified individual or group.

- [Sender](#)

This section is where you configure the pattern recognition to determine what represents who or what sent the messages. This section can also be set to a static sender to indicate that all traffic processed from the serial line should declare that it is from the specified value.

- [Message](#)

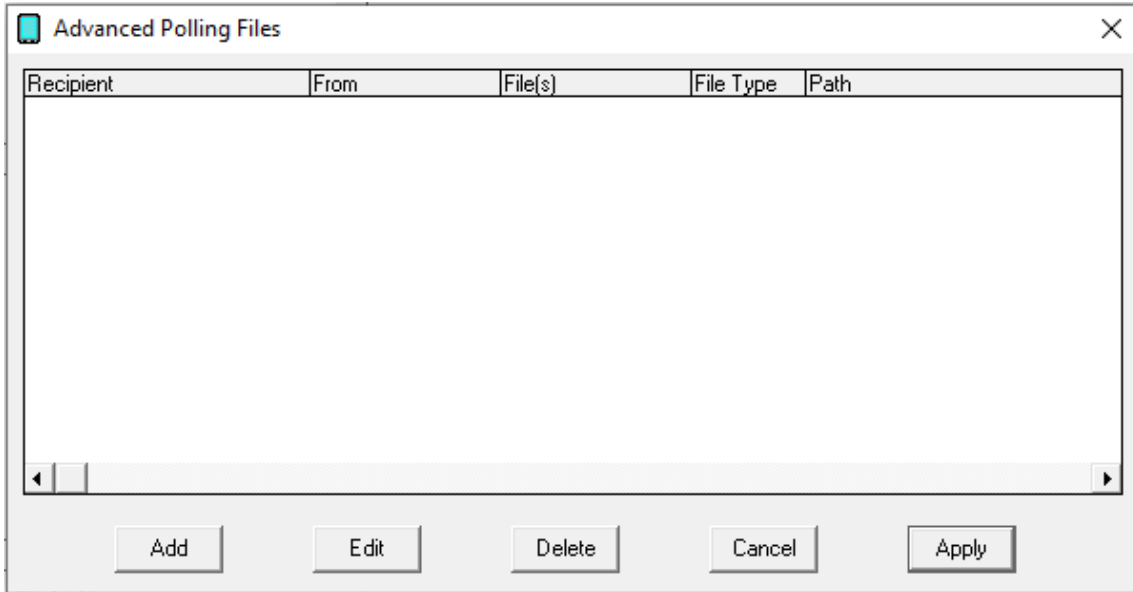
This section is where you configure the pattern recognition to determine what represents the body of the message to be delivered. This section can also be set to a static message to provide the message text to be delivered.

- [Advanced Polling Example](#)

This section shows an example of how to implement the GetASCII module's Advanced Polling to process XML, HTML and similar ascii formatted languages.

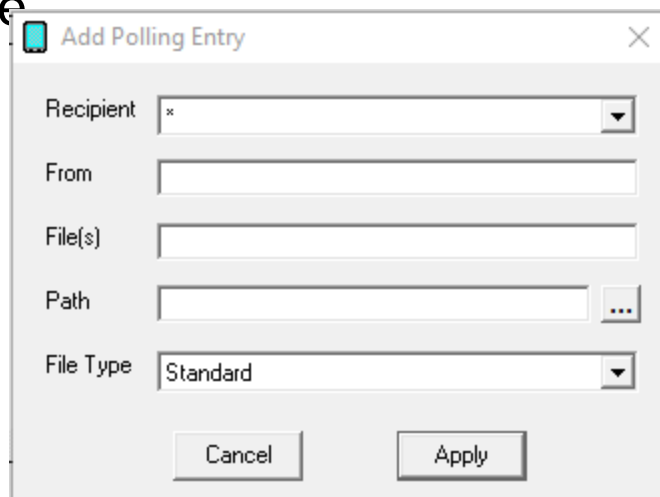
PageGate

This section determines the directory or series of directories that GetASCII should look for content within and also specifies style of implementation as well as the naming convention of the files being worked with.



Option	Function
Add	Adds a new Advanced Polling rule.
Edit	Edits an existing Advanced Polling rule
Delete	Removes an existing Advanced Polling rule.
Cancel	Cancel all recent changes.
Apply	Applies new changes and rules.

To add an Advanced Polling Files option, click on the Add button.



Setting	Function
Recipient	<p>By default, this field is set to all recipients (*).</p> <p>If the asterisk(*) option is used, this tells GetASCII that the recipient information will be contained within the file(s) processed.</p> <p>If you select a recipient or group from the list, all messages processed by this rule will be delivered solely to the selected recipient or group.</p>
From	<p>This field is only used when a specific recipient or group is specified in the Recipient field, otherwise, leave this field blank.</p> <p>If a specific recipient is used in the Recipient field, the 'From' field specifies what sender value to use for those messages.</p>
File(s)	<p>This field determines the naming convention of the file(s) you want GetASCII to look for. Wildcard characters (* and ?) may be used.</p> <p>For example, an entry of *.txt will poll all files that end with .txt</p> <p>An entry of *.?at will collect files like cat, bat, and rat, but not 'flat'.</p> <p>An entry of custom*.xml will collect files like custom1.xml and custom_naming_scheme.xml or customer1234.xml</p>
Path	<p>This field determines the directory that this Advanced Polling entry will monitor. This can be set to a local path (example: c:\PageGateData\ASCII), UNC path (example: \\server\path\), or mapped drive location (example: S:\CADOutputLocation\).</p> <p>Wildcard characters are not valid in this field.</p>

	<p>If you're running PageGate as a Windows Service and your Advanced Polling is referencing a UNC path, you will usually need to provide a service account to the PageGate or PageGate GetASCII Windows services.</p>
File Type	<p>There are four options for the file type of each Advanced Polling rule:</p> <ul style="list-style-type: none">• <u>Standard</u> Selecting this option tells GetASCII that text file(s) being referenced are in the standard ascii format. This option is usually used when files are in the standard ascii format, but aren't named with the standard .asc extension or are written to a directory other than the basic polling directory. When the Standard file type is selected, the other Advanced Polling sections (Symbols, Record, Recipient, Sender, and Message) are not referenced in the file read process.• <u>Custom</u> When the Custom file type is selected, the other Advanced Polling options (Symbols, Record, Recipient, Sender, and Message) are used. Each text file must contain a single message, but the message information is not in the regular format and must be extracted from the text file using the Advanced Polling configuration.• <u>Adding</u> The ascii file specified by the File(s) field periodically has message information added to the end of the file. With this file type the GetASCII Interface will read in any new data added to the file and extract the message information using the Symbols, Record, Recipient, Sender, and Message options.• <u>Changing</u> The ascii file changes periodically. Each time the file specified in the File(s) field is changed, the GetASCII Interface will attempt to extract message information from it using the Symbols, Record, Recipient, Sender, and Message options. This option is different from Adding because the Adding option only looks at the end of the file specified. The Changing option assumes the entire file will be modified, not appended to.

PageGate Admin: How to Configure

6

PageGate

Symbols

This section determines certain special characters and their functions.

Single Character Wildcard	<input type="text" value="?"/>
Multiple Character Wildcard	<input type="text" value="*"/>
End Of Record Symbol	<input type="text" value="~"/>
Ascii Value Delimiter	<input type="text" value="\"/>

PageGate

Setting	Function
Single Character Wildcard	<p>When scanning through the ascii data, this character can be used to stand in for any single character.</p> <p>For example, let's say you use Fire?</p> <p>That would tell the GetASCII module to recognize any word that began with Fire and had one extra letter on the end. So, it would recognize Fires or Fired but would not recognize Fire alone.</p> <p>As another example, let's say you use ??/??/????</p> <p>That would tell GetSerial to recognize any date stamp in the two digit month, two digit day, four digit year configuration.</p>
Multiple Character Wildcard	<p>When scanning through the ascii data, this character can be used to stand in for any character or group of characters.</p> <p>For example, let's say you use Fire*</p> <p>That would tell the GetASCII module to recognize any instance of the word Fire, regardless of whether there are any characters after the word. So, it would recognize Fire, Fired, Fires, Firehouse or Fireman and it would also recognize 'Fire' in phrases like Fire Sale or Fire Department.</p> <p>As another example, let's say you use *age*</p> <p>That would tell GetSerial to recognize any reference to the three letters age close to each other, regardless of whether that was in the word PageGate or NotePage or ageless or damage. Any instance of the three characters together will be recognized.</p>
End of Record Symbol	<p>This character stands for the end of record symbol.</p>
ASCII Value Delimiter	<p>This symbol is used to help represent non-printable characters that normally can't be typed like horizontal tabs, carriage returns and line feeds. This character is always followed three numbers which represent the decimal value for a character.</p> <p>For example, this represents a carriage-return: \013</p> <p>For a full list of ascii codes, visit http://www.asciitable.com/</p>
Apply	<p>Applies all changes made to the Symbols.</p>

Record

This section is where you configure the pattern recognition to determine what represents the start and end of a message sequence.

PageGate Admin: How to Configure PageGate

The screenshot shows a configuration window with the following fields:

- Start Pattern: * (text input)
- Start Offset: 0 (numeric input)
- End Pattern: ~ (text input)
- End Offset: 0 (numeric input)
- Apply (button)

- **Start Pattern**

This value determines the start of relevant information. For example, let's say this is the contents of a file you need to process:

```
SubscriberName=AgencyToDispatch
[MESSAGE]
Line1=Type of call
Line2=Address information.
Line3=Cross-street information.
Line4=Dispatcher notes.
```

You would want to use * as the Start Pattern as we want GetASCII to recognize that the relevant data begins at the very first character of the file.

- **Start Offset**

This value determines how many characters before or after the identified start pattern to begin character processing. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will begin the data record 5 characters after the first character in the Start Pattern. A value of -5 means that GetSerial will begin the data record 5 characters before the first character the Start Pattern.

Using our example, you would set the Start Offset to 0 to indicate that the relevant data begins at the very first character read in.

Note: the value -999 has special meaning, and is typically used in conjunction with the use of the Multiple Character Wildcard in the search strings.

If -999 is used as a Start Offset, an offset of one character past the string that matches the start pattern, is used. This will cause the number of characters that were found to match the Start Pattern to be skipped, even though the actual number of characters in the Start Pattern matching string aren't known ahead of time.

- **End Pattern**

PageGate Admin: How to Configure

PageGate

This value determines the end of relevant information. For example, let's say this is the contents of a file you need to process:

```
SubscriberName=AgencyToDispatch
[MESSAGE]
Line1=Type of call
Line2=Address information.
Line3=Cross-street information.
Line4=Dispatcher notes.
```

You would want to use ~ as the End Pattern as that tells PageGate that the very last character in the file is the end of relevant information.

- **End Offset**

This value determines how many characters before or after the identified end pattern to end character processing. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will end the data record 5 characters after the last character in the End Pattern. A value of -5 means that GetSerial will end the data record 5 characters after the last character the End Pattern. To use our earlier example, if we don't want the word END to be a part of the record processed, we would set the Start Offset value to -3. That way, the data record ends three characters before the last character of END.

Note: the value -999 has special meaning, and is typically used in conjunction with the use of the Multiple Character Wildcard in the search strings.

If -999 is used as an End Offset, an offset of one character before the string that matches the end pattern, is used. This will cause the number of characters that were found to match the End Pattern to be skipped, even though the actual number of characters in the End Pattern matching string aren't known ahead of time.

Recipient

<u>IMPORTANT</u>	
\	<p>This character, unless specified otherwise in the Symbols section, represents the character that tells PageGate that the three characters following it represent an ascii decimal value/ascii code. For instance, if you want to tell PageGate to look for a carriage return, you would use a backslash and the three character ascii decimal value representing a carriage return. Written out, that would be \013</p> <p style="text-align: center;">For a full list of ascii codes, visit http://www.asciitable.com/</p>

This section is where you tell GetASCII what represents who the messages should go to.

PageGate Admin: How to Configure PageGate

The screenshot shows the configuration interface for PageGate. The 'Dynamic Recipient' option is selected with a radio button. The 'Recipient' field is a dropdown menu. The 'Start Pattern' field contains an asterisk (*), 'Start Offset' is 0, 'End Pattern' contains a tilde (~), and 'End Offset' is 0. There is an unchecked checkbox for 'Use Lookup Table' and an 'Edit Table' button below it. An 'Apply' button is at the bottom.

There are two possible configurations you can use:

- **Static Recipient**

The screenshot shows the configuration interface with the 'Static Recipient' option selected. The 'Recipient' field is a dropdown menu with a mouse cursor pointing to it. The 'Dynamic Recipient' option is unselected. The 'Start Pattern' field contains an asterisk (*).

Selecting this option tells the GetASCII module that all messages processed from the ascii file should go to the specified recipient or group.

- **Dynamic Recipient**

The screenshot shows the configuration interface with the 'Dynamic Recipient' option selected. The 'Start Pattern' field contains an asterisk (*), 'Start Offset' is 0, 'End Pattern' contains a tilde (~), and 'End Offset' is 0. There is an unchecked checkbox for 'Use Lookup Table' and an 'Edit Table' button below it.

Selecting this option tells the GetASCII module to look for specific patterns that denote where the recipient information is located in within the [record](#).

- **Start Pattern**

PageGate

This value determines the first character in the recipient value.

For example, let's say this is the [record](#) identified and let's also say that who the message needs to go to is contained in the SubscriberName= field:

SubscriberName=AgencyToDispatch

[MESSAGE]

Line1=Type of call

Line2=Address information.

Line3=Cross-street information.

Line4=Dispatcher notes.

You would want to use SubscriberName= as the Start Pattern of the recipient information.

Note: Start patterns are included in the recipient processed unless you specify the otherwise with the Start Offset. See below.

• **Start Offset**

This value determines how many characters before or after the identified start pattern to accept as the recipient value. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will begin looking for the recipient value 5 characters after the first character in the Start Pattern. A value of -5 means that GetSerial will begin looking for the recipient value 5 characters before the first character the Start Pattern.

To use our earlier example, we would want a Start Offset of 15 because there are 15 characters in SubscriberName= and we don't want that text included as part of the recipient or group designation.

• **End Pattern**

This value determines the last character in the recipient value.

Using our earlier example, we've already specified that the recipient value starts with SubscriberName= but what does it end with?

SubscriberName=AgencyToDispatch

[MESSAGE]

Line1=Type of call

Line2=Address information.

Line3=Cross-street information.

Line4=Dispatcher notes.

This is where things are ultimately up to your preferences. The first thing that technically follows the recipient name in our example is a carriage return followed by a line feed (represented visually as a line break but as two ascii codes/characters in the text being

PageGate Admin: How to Configure PageGate

6

processed), not the text [MESSAGE]. However, you could tell PageGate to use either \013 (the ascii value representing a carriage return) or [MESSAGE] as the End Pattern as both clearly identify the end of recipient information. However, you would use a different End Offset for each. See below.

- **End Offset**

This value determines how many characters before or after the identified End Pattern to stop character recognition. The discovered value is not included in the overall statement. This field can be set to positive or negative values.

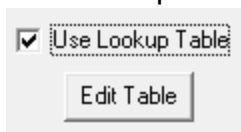
For instance, a value of 5 means GetASCII will stop looking for the recipient value 5 characters after the first character in the End Pattern. Using our earlier example, if your End Pattern is [MESSAGE] and you specify a value of 5, the data for the recipient will end with [MESS and you don't want that. If your End Pattern is \013 and you specify a value of 5, the data for the recipient will end with <CR><LF>[ME and you don't want that.

On the other side of the coin, a value of -5 means GetASCII will stop looking for the recipient value 5 characters before the first character the End Pattern. Using our earlier example, if your end pattern is either \013 or [MESSAGE] and you specify a value of -5, the first five characters in the recipient line will be lost and you don't want that.

To use our earlier example, if we have \013 specified as the End Pattern, you would use a End Offset of -1 thereby telling GetASCII not to include the carriage return as part of the recipient or group designation. If we have [MESSAGE] specified as the End Pattern, we would use an End Offset of 0.

- **Lookup Table**

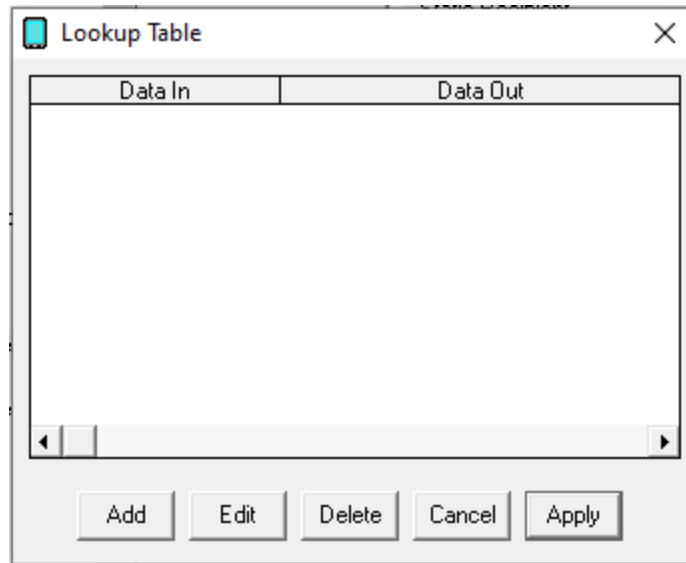
The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed. To enable this feature, check Use Lookup Table and click Edit Table.



This feature enables PageGate to perform basic search and replace commands on values read in as the Recipient, which allows you to interchange or translate values as needed.

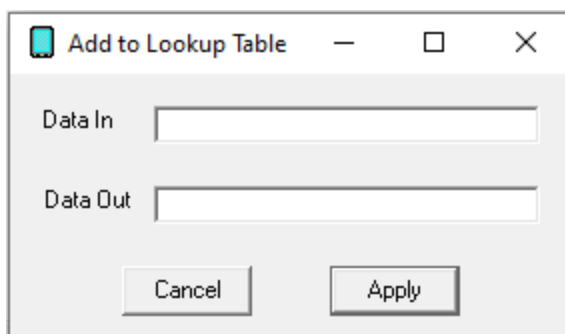
PageGate Admin: How to Configure PageGate

6



Button	Function
Add	Adds a new search and replace value.
Edit	Edits an existing search and replace value.
Delete	Removes an existing search and replace value.
Cancel	Closes the table without saving changes.
Apply	Applies updates made to the table.

When adding new entries to the table, you'll be presented with this dialog:



- **Data In**

This value represents what you want to search for. Wildcard characters can be used.

- **Data Out**

This value represents what you want to replace the text with.

For example, if we had a 'Data In' value of oranges and a 'Data Out' value of bananas, the program would replace any instance of the word oranges with the word bananas.

PageGate Admin: How to Configure PageGate

6

To provide a more practical example, let's use the [example from the GetASCII - Settings - Recipient section](#) of the documentation and say that we needed to translate AgencyToDispatch in to recipient or group name as you've entered it in to the PageGate Admin. You could use a 'Data In' value of AgencyToDispatch and a 'Data Out' value of Station14 so any messages that came addressed to the literal text AgencyToDispatch were routed to the group called Station14 in PageGate.

Sender

IMPORTANT
<p>This character, unless specified otherwise in the Symbols section, represents the character that tells PageGate that the three characters following it represent an ascii decimal value/ascii code. For instance, if you want to tell PageGate to look for a carriage return, you would use a backslash and the three character ascii decimal value representing a carriage return. Written out, that would be \013</p>
<p>For a full list of ascii codes, visit http://www.asciitable.com/</p>

This section is where you tell GetASCII who or what sent the messages.

Static Sender
Sender

Dynamic Sender
Start Pattern
Start Offset
End Pattern
End Offset
 Use Lookup Table

There are two possible configurations you can use:

- Static Sender

Static Sender
Sender

Selecting this option tells the GetASCII module that all messages processed should identify themselves as having been sent from the specified value. This is the value used in the majority of GetASCII implementations.

PageGate Admin: How to Configure PageGate

- Dynamic Sender

The screenshot shows a configuration panel for 'Dynamic Sender'. It contains four input fields: 'Start Pattern' with an asterisk, 'Start Offset' with the value 0, 'End Pattern' with a tilde, and 'End Offset' with the value 0.

Selecting this option tells the GetASCII module to look for specific patterns that denote where the sender information is located in within the [record](#).

- **Start Pattern**

This value determines the first character in the sender value.

- **Start Offset**

This value determines how many characters before or after the identified start pattern to accept as the sender value. This field can be set to positive or negative values.

For instance, a value of 5 means that GetASCII will begin looking for the sender value 5 characters after the first character in the Start Pattern. A value of -5 means that GetASCII will begin looking for the sender value 5 characters before the first character the Start Pattern.

- **End Pattern**

This value determines the last character in the sender value.

- **End Offset**

This value determines how many characters before or after the identified End Pattern to stop character recognition. This field can be set to positive or negative values.

For instance, a value of 5 means that GetASCII will stop looking for the sender value 5 characters after the first character in the End Pattern. A value of -5 means that GetASCII will stop looking for the sender value 5 characters before the first character the End Pattern.

- **Lookup Table**

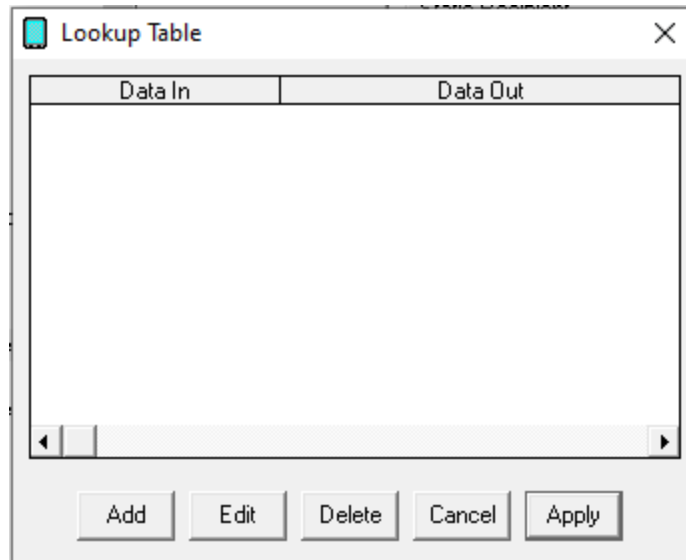
The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed. To enable this feature, check Use Lookup Table and click Edit Table.

The screenshot shows a checkbox labeled 'Use Lookup Table' which is checked, and a button labeled 'Edit Table' below it.

PageGate Admin: How to Configure PageGate

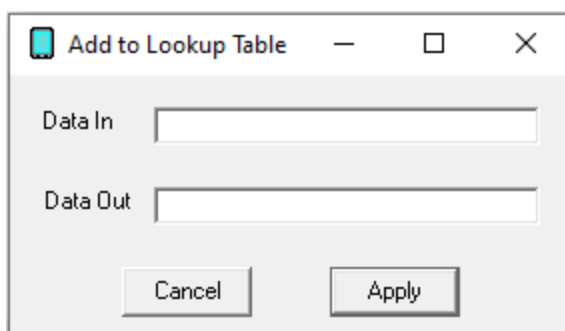
6

This feature enables PageGate to perform basic search and replace commands on values read in as the Sender, which allows you to interchange or translate values as needed.



Button	Function
Add	Adds a new search and replace value.
Edit	Edits an existing search and replace value.
Delete	Removes an existing search and replace value.
Cancel	Closes the table without saving changes.
Apply	Applies updates made to the table.

When adding new entries to the table, you'll be presented with this dialog:



- **Data In**

This value represents what you want to search for. Wildcard characters can be used.

- **Data Out**

PageGate Admin: How to Configure PageGate

6

This value represents what you want to replace the text with.

For example, if we had a 'Data In' value of oranges and a 'Data Out' value of bananas, the program would replace any instance of the word oranges with the word bananas.

To provide a more practical example, let's say we wanted to translate A834 in to Dispatch Center. You would use a 'Data In' value of A834 and a 'Data Out' value of Dispatch Center.
Message

<u>IMPORTANT</u>
This character, unless specified otherwise in the Symbols section, represents the character that tells PageGate that the three characters following it represent an ascii decimal value/ascii code. For instance, if you want to tell PageGate to look for a carriage return, you would use a backslash and the three character ascii decimal value representing a carriage return. Written out, that would be \013
For a full list of ascii codes, visit http://www.asciitable.com/

This section is where you tell GetASCII what represents the message.

Static Message
Message

Dynamic Message
Start Pattern
Start Offset
End Pattern
End Offset

Use Lookup Table

There are two possible configurations you can use:

- Static Message

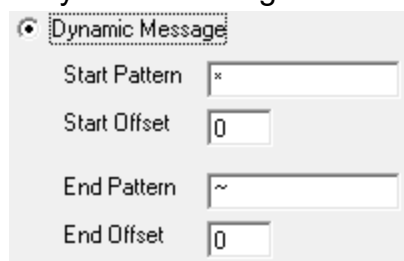
Static Message
Message

Selecting this option allows you to specify a pre-canned message when a file is read in. This is useful for configuring a system so that any time a file appears in the folder

PageGate

GetASCII is monitoring, a pre-canned message is sent to the necessary recipients and groups.

- **Dynamic Message**



Selecting this option tells the GetASCII module to look for specific patterns that denote where the message is located in within the [record](#).

- **Start Pattern**

This value determines the first character in the message value.

To use [the example we've been working with](#), the record originally started as:

```
SubscriberName=AgencyToDispatch  
[MESSAGE]  
Line1=Type of call  
Line2=Address information.  
Line3=Cross-street information.  
Line4=Dispatcher notes.
```

We then configured GetASCII to treat SubscriberName=<value> as the recipient line, which removes it from the data string. So, by the time the Message processing happens, this is the remaining data:

```
[MESSAGE]  
Line1=Type of call  
Line2=Address information.  
Line3=Cross-street information.  
Line4=Dispatcher notes.
```

As a result, we would want a Start Pattern of Line1= to indicate that the message begins with that text. This effectively removes the [MESSAGE] line as irrelevant data.

- **Start Offset**

This value determines how many characters before or after the identified start pattern to accept as the message. This field can be set to positive or negative values.

For instance, a value of 5 means that GetASCII will begin looking for the message 5 characters after the first character in the Start Pattern. A value of -5 means that GetASCII will begin looking for the message 5 characters before the first character the Start Pattern.

PageGate Admin: How to Configure

PageGate

Using our example, you would want to set the Start Offset to 6 because there are 6 characters in Line1=

- **End Pattern**

This value determines the last character in the message.

In our example, you would want to set the End Pattern to the [End of Record symbol](#) in the [GetASCII - Settings - Symbols](#) section. This tells PageGate that the last character of the data line is the termination point for the message. So, by configuring a Start Pattern of Line1= and ~ for the End Pattern, we're telling PageGate that all of the remaining relevant data is between Line1= and the end of the file.

- **End Offset**

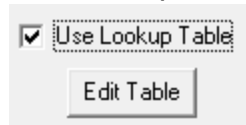
This value determines how many characters before or after the identified End Pattern to stop character recognition. This field can be set to positive or negative values.

For instance, a value of 5 means that GetASCII will stop looking for the message 5 characters after the first character in the End Pattern. A value of -5 means that GetASCII will stop looking for the message 5 characters before the first character the End Pattern.

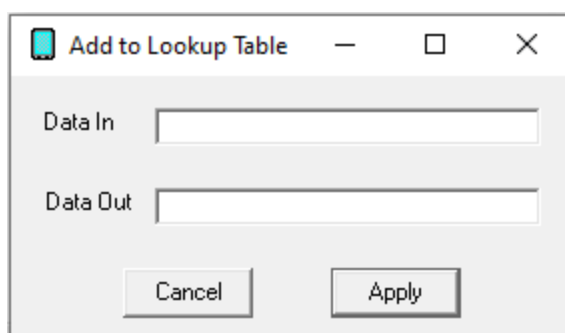
In our example, you would set the End Offset to 0 to indicate that the last character of the file was the last relevant character to process.

- **Lookup Table**

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed. To enable this feature, check Use Lookup Table and click Edit Table.



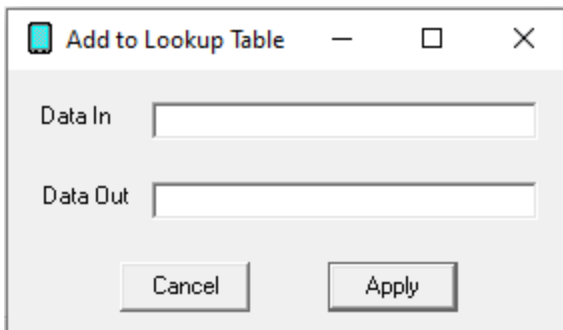
This feature enables PageGate to perform basic search and replace commands on values in the body of the message, which allows you to interchange or translate values as needed.



PageGate Admin: How to Configure PageGate

Button	Function
Add	Adds a new search and replace value.
Edit	Edits an existing search and replace value.
Delete	Removes an existing search and replace value.
Cancel	Closes the table without saving changes.
Apply	Applies changes made to the table.

When adding new entries to the table, you'll be presented with this dialog:



- **Data In**

This value represents what you want to search for. Wildcard characters can be used.

- **Data Out**

This value represents what you want to replace the text with.

For example, if we had a 'Data In' value of oranges and a 'Data Out' value of bananas, the program would replace any instance of the word oranges with the word bananas.

To provide a more practical example, let's use the [example from the GetASCII - Settings - Message section](#) of the documentation. This is the message being processed:

Type of call

Line2=Address information.

Line3=Cross-street information.

Line4=Dispatcher notes.

Assuming that * is your [multiple character wildcard](#), to remove the text Line2=, Line3=, Line4=, etc from the body of the message, you would use the following:

Data In: *Line2=*

Data Out: Leave this field completely empty.

Data In: *Line3=*

Data Out: Leave this field completely empty.

PageGate Admin: How to Configure PageGate

6

and so on.

Advanced Polling Example

PageGate's GetASCII module can be configured to read any ascii formatted content, regardless of whether that content is a stream of text or a formatted language like HTML, XML, Java, etc. In this example, you'll be going through how to configure PageGate to read XML content but you can use these steps just as easily for HTML, JSON or another ascii text formatted language by replacing the XML flags with the appropriate language's flags/tags/statements/etc.

First, we'll need to create a basic polling directory and get the module online:

- 1) In Windows, create a sub-directory of your PageGateData folder and name it ASCII
Note: This is arbitrary and you can name the folder whatever you like). For example, if you're using PageGate's default directories, you would create C:\PageGateData\ASCII
- 2) Open the PG Admin.
- 3) Go to Interfaces - GetASCII - Settings.
- 4) Set the "Polling Directory" to the ASCII folder you created in step 1. By default, C:\PageGateData\ASCII
- 5) Check Enabled.
- 6) Click Apply.
- 7) When prompted, you do want to enable this for all existing groups and recipients.
- 8) Go to Program - Settings.
- 9) In "Run on this server", check GetASCII
- 10) Click Apply.
- 11) When prompted for credentials, please reference the [Windows Credentials](#) section of the [Run Style](#) documentation.
- 12) When "Switching Run Styles" goes away, open the Windows Services list

Next, we'll need to configure the GetASCII module to process your XML content:

- 1) Go to Interfaces - GetASCII - Settings - Advanced Polling - Files.
- 2) Click Add.
- 3) Specify the following:
Recipient: *

PageGate Admin: How to Configure PageGate

From: Enter a sender's value. CAD, for example.

File(s): If you want the program to read all XML files from the folder, use *.xml

If you want the program to only read for a specific file name, use filename.xml

Combinations of the two are allowed. For example, you could use 2017*.xml and the program would only look for files that began with 2017 and had an extension of xml.

Path: Enter the directory the .xml files will be in.

File Type: Custom

4) Click Apply.

5) Click Apply.

6) Go to Interfaces - GetASCII - Settings - Advanced Polling - Record

7) Specify the following:

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

General Notes: This effectively tells PageGate that the data to be processed is the first character in the file until the last character of the file.

8) Click Apply.

9) Go to Interfaces - GetASCII - Settings - Advanced Polling - Recipient.

10) If you want all of the processed content always delivered to the same recipient or group, select Static Recipient and select the recipient or group to which these messages should be delivered.

If there is an XML tag that denotes who these messages should go to, select Dynamic Recipient and specify the following:

Start Pattern: <YourRecipientXMLTag>

Start Offset: Set this to the number of characters contained in your xml tag. For example, if your XML tag is <Recipient>, you would set Start Offset to 11 as there are 11 characters in <Recipient>.

End Pattern: </YourRecipientXMLTag>

End Offset: 0

11) Click Apply.

12) Go to Interfaces - GetASCII - Settings - Advanced Polling - Sender.

PageGate Admin: How to Configure

PageGate

- 13) Select Static Sender and enter a value in to the Sender's Name field. For example, you could use CAD.
- 14) Click Apply.
- 15) Go to Interfaces - GetASCII - Settings - Advanced Polling - Message.
- 16) Select Dynamic Message and specify the following:

If the entire message is contained within a single XML tag, you would specify the following:

Start Pattern: <XMLMessageTag>

Start Offset: 0

End Pattern: </XMLMessageTag0>

End Offset: 0

For example, let's say that your message is contained in XML content like this:

```
<Message>The entire body of the message is contained in this field.</Message>
```

You would use the following:

Start Pattern: <Message>

Start Offset: 0

End Pattern: </Message>

End Offset: 0

As another example, let's say your message is contained in XML content like this:

```
<Message>
<line1>Relevant data</line1>
<line2>More relevant data</line2>
<irrelevant_data>Value</irrelevant_data>
<irrelevant_data>Value</irrelevant_data>
<line3>More relevant data</line3>
</Message>
```

You would use this:

Start Pattern: <Message>

Start Offset: 0

End Pattern: </Message>

End Offset: 0

If the message is contained across multiple XML tags with no over-arching header tag, you would specify the following:

PageGate Admin: How to Configure PageGate

Start Pattern: <FirstRelevantTag>
Start Offset: 0

End Pattern: </LastRelevantTag>
End Offset: 0

For example, let's say your message is contained in XML content like this:

```
<message_line_1>First line of the message</message_line_1>
<irrelevant_data>Value</irrelevant_data>
<irrelevant_data>Value</irrelevant_data>
<message_line_2>Second line of the message</message_line_2>
<message_line_3>Third line of the message</message_line_3>
```

You would use the following:

Start Pattern: <message_line_1>
Start Offset: 0

End Pattern: </message_line_3>
End Offset: 0

17) Check "Use Lookup Table". This function will allow us to remove the raw text XML tags from the text to be delivered.

18) Click Edit Table.

19) Click Add.

General Notes: The Data In value represents the data you want either removed or modified. If you want the data removed, leave the Data Out field completely empty.

For example, if you wanted to change every instance of the word pecans to the word oranges, you would use this:

Data In: *pecans*

Data Out: oranges

However, if you wanted to remove every instance of the word pecans, you would use this:

Data In: *pecans*

Data Out:

To make this a bit more relevant for XML content, however, let's say we wanted to completely remove an entire XML statement as it didn't contain any relevant data. We would use this:

Data In: *<value>*</value>*

Data Out:

PageGate Admin: How to Configure PageGate

With that, we're telling PageGate to completely remove the entire XML statement from the text processed. For example, `*<DATE>*</DATE>*` tells PageGate that there isn't any relevant data in the `<DATE>` tag and to simply remove the entire section from what needs to be delivered.

As another example, if you want the data in the `<ServerName>` tag but you don't want the actual text `<ServerName>` to appear in the text delivered, you would use this:

Data In: `*<ServerName>*`
Data Out:

and you would want to add an additional entry to remove the `</ServerName>` closing tag:

Data In: `*</ServerName>*`
Data Out:

Just to note, anything expressed as a `\XXX` value is an ASCII code (<http://www.asciitable.com/>). For example, `\009` is a tab, `\013` is a carriage return and `\010` is a line feed.

For example, if you want PageGate to replace all instances of three carriage return/line feed statements in a row and with a single carriage return/line feed, you would use this:

Data In: `*\013\010\013\010\013\010*`
Data Out: `\013\010`

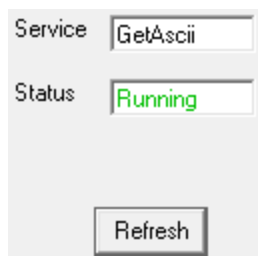
As another example, if you want to remove all tabs from the data, you would use this:
Data In: `*\009*`
Data Out:

20) After you've added the necessary entries, click on Apply.

21) Click Apply.

Status

This section displays the current status of the GetASCII module.



PageGate Admin: How to Configure PageGate

If the status of GetASCII is stopped, it indicates that the module is not running and this can be caused by a few things.

- GetASCII will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- Enabled must be checked in the [settings of the GetASCII module](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

Log

This log is a record of information and traffic processed by the GetASCII module.



PageGate

Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

Command Line Interface

The Command Line Interface function of GetASCII can be used to send messages to PageGate from:

Command Line	Messages can be sent from a command prompt.
Windows Command Line	Messages can be sent from the Windows Run menu.
Third Party Applications	Messages can be sent from any application that can issue a command line command (shell command).
Custom Applications	This is a popular interface for developers who want an easy way to add messaging to their application.

The most basic components of the Command Line interface are the `sendpage.exe` and `sendpage32.exe` files. By default, these files are hosted in the [Polling Directory](#) specified in the [base settings](#) of the GetASCII module but you can copy these files to any location to have them function and output at that location.

The difference between the `sendpage.exe` and `sendpage32.exe` files is that the `sendpage.exe` is intended to be run from a DOS Command Line or by 16-bit applications and `sendpage32.exe` is a 32-bit program, which is what most Windows based applications will use.

To pass a message to `sendpage.exe` or `sendpage32.exe`, use this command structure:

```
sendpage.exe <recipient> <sender> <message>
```

PageGate Admin: How to Configure PageGate

<recipient> should be replaced with the intended recipient or group's name as it exists in the PageGate Admin. This value keys specifically off of the Recipient field for recipients and the Group field for groups. This is a required value.

<sender> should be replaced with who or what is sending the message. This is a required value.

<message> should be replaced with what you want the body of the message to say.

Ad-Hoc with the Command Line interface

It's also important to note that you can use [Ad-Hoc recipients](#) with the command line interface. When using the sendpage.exe or sendpage32.exe command line executables to pass messaging parameters to PageGate, this is how to call on an ad-hoc recipient: <location of files>\sendpage32.exe <ad-hoc recipient name>;<value>;<value>;<etc> <sender> <message>

For example, if you have the sendpage32.exe hosted in c:\PageGateData\ASCII\ and your [ad-hoc recipient's](#) name is 'ah', then you sent the following command:
c:\PageGateData\ASCII\sendpage32.exe ah;7819999998;8290500234 CAD This is a test message

That would send a message to the phone numbers specified from the sender 'CAD' and the message would say 'This is a test message'.

GetMail

PageGate has a fully functional mail server engine as a core part of the program but without the GetMail module, you only ever access the outbound functions of this mail server. The GetMail module unlocks the receiving portion of our mail server engine and a few features associated with the mail server in general. This allows you to receive SMTP/Email traffic and translate it in to outbound messages in a different format. For example, you could email a message to PageGate to deliver as a message to pagers, cell phones and/or other email addresses.

Enabling this interface will allow you to access the [Email sub-section](#) of the recipients and groups in the PageGate Admin. This allows you to have all traffic to a recipient or group forwarded to the specified email address as well as allowing PageGate to host a POP accessible mailbox on the PageGate server for this recipient or group.

As mentioned above, GetMail allows you to receive SMTP/Email traffic and translate it in to outbound messages in a different format and it can receive email/SMTP messages in one of two ways:

- **Sub-domain hosting**

PageGate

The best method of email integration is to have PageGate host a sub-domain off of your existing domain. To do this, you'll want to talk to your mail server administrator and ask them to create this sub-domain for you, then redirect the MX record for that sub-domain to the internal IP address of the PageGate server. For example, let's say you have an email domain of example.com. So, you might have something.example.com or messaging.example.com or pg.example.com created and redirected to the internal IP of the PageGate server. At that point, PageGate 'owns' the sub-domain and processes all traffic intended for it.

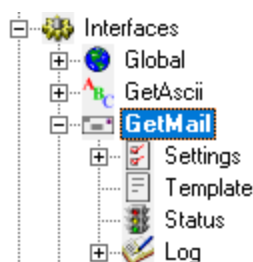
So, let's say that we have a group called sales_team and we have GetMail configured to use messaging.example.com. You could send an email to sales_team@messaging.example.com and that email would be delivered to all members of the group as it exists in PageGate.

• POP Mailbox

POP is a relatively antiquated method of receiving email and many modern mail servers don't support it. However, if your mail server does still support POP mailboxes, you can configure GetMail to periodically check those email addresses, then read in any email waiting in the mailbox and deliver that traffic to a recipient or group of your specification.

If your POP server requires SSL/TLS authentication, be sure to set the Pop_Delivery_Port Template value to 995. For more information, please see the [Parameters for GetMail](#) section.

With the ubiquity of email, more and more modern applications are integrating the ability to send email. If your application has the ability to send email, it can easily be integrated with PageGate by using the GetMail module. It is worth noting that this interface can be used to integrate applications running on different platforms (Unix, Mac, etc.).



• [Settings](#)

This section controls the implementation of the GetMail API.

• [Status](#)

This section displays the running/not running status of the interface.

• [Template](#)

This section allows you to configure a custom template for messages processed by this API.

PageGate Admin: How to Configure PageGate

- [Log](#)

This section contains a log of all activity recorded by the PageGate Mail Server and GetMail module.

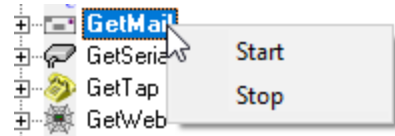
Right clicking on GetMail will provide a context menu with two options.

- [Start](#)

Selecting this option will start the GetMail module and PageGate Mail Server service, if stopped.

- [Stop](#)

Selecting this option will stop the GetMail module and PageGate Mail Server service, if started.



Settings

Enabled

Local Domain

Postmaster

Relay All Mail

Relay Server

Use system's DNS servers

Set static DNS servers

DNS Server 1

DNS Server 2

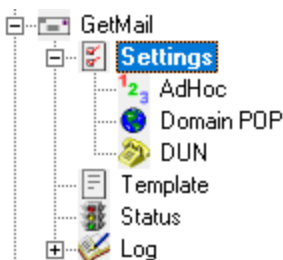
POP Interval (minutes)

Setting	Function
Enabled	When the Enabled checkbox is checked, it tells the PageGate server to run the GetMail module.
Local Domain	This is a multi-purpose field. If the GetMail module is only being configured to send or check email, this field needs to be the sending domain. For example, if your email address was support@notepage.com , you would put notepage.com in the Local Domain as that is the domain the messages are being sent from. If the GetMail module is being configured to function as a POP3 server, the Local Domain needs to be set to the domain used by PageGate when receiving email. This is an internet domain name, not a Microsoft networking domain. Only email with an address in this domain will be accepted by

PageGate

	PageGate. This setting must be a valid domain name for to work properly (Note: an IP address can be used here instead of a domain name, if email is addressed to match. This is commonly done during testing when a domain name has not yet been assigned to this machine).
Postmaster	The Postmaster setting needs to be set to a valid email address. This email address is sent all warnings, alerts, and bad messages generated by the GetMail module. Leaving this field blank will drop all messages that would normally go to the postmaster. The Postmaster setting is also used when no valid Sender's Information is present in an outbound email message.
Relay All Mail	This option should only be checked if port 25 is not open on your network. To determine whether port 25 is open or closed on your network, please run a Port Scan .
Relay Server	This field allows you to specify a relay server for SMTP messages to be routed through. This option should only be used if port 25 is not open on your network. To determine whether port 25 is open or closed on your network, please run a Port Scan .
DNS Server 1	This field allows you to specify the primary DNS server for SMTP delivery and is only valid if port 25 is open on your network. To determine whether port 25 is open or closed on your network, please run a Port Scan .
DNS Server 2	This field allows you to specify the secondary DNS server for SMTP delivery and is only valid if port 25 is open on your network. To determine whether port 25 is open or closed on your network, please run a Port Scan .
POP Interval	If any of the GetMail Module's POP functions are used, this setting determines how often (in minutes) their processes occur.

Expanding Settings will display the following menu:



- [Ad-Hoc](#)

This section configures PageGate's mail server to accept certain values to pass to an Ad-Hoc recipient.

- [Domain POP](#)

A relatively antiquated but still supported method of mail collection for an entire email domain.

PageGate Admin: How to Configure

PageGate

A relatively antiquated but still supported method of mail collection involving a Dial Up Network.

AdHoc

This section controls how GetMail implements AdHoc Email. The interface can handle Ad-Hoc email in two ways, both of which can be used separately or together.

- [The first method](#) is to accept email addressed to the ID/PIN number that the message should use for delivery (i.e. [1234567@pagingserver.com](#)).
- [The second method](#) is to have the Ad-Hoc Email function accept incoming email with the ID/PIN numbers of the intended destination device on the subject line.

Button	Function
Enable Ad-Hoc Mail	Enables the use of Ad-Hoc ranges for email processing.
Add	Adds a new Ad-Hoc range.
Edit	Edits an existing Ad-Hoc range.
Delete	Removes an existing Ad-Hoc range
Apply	Applies updates made to the table.

Method 1

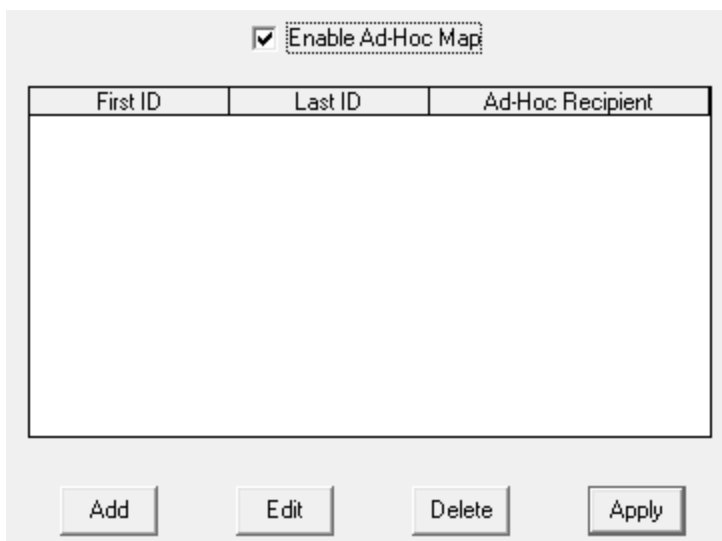
To use this method with email, simply address an email message to the PageGate Server's [Local Domain](#) with an ID/PIN number (within the range set in steps 6 and 7 below) as the part of the email address before the @ (i.e. [1234567@mypageserver.com](#)).

PageGate Admin: How to Configure PageGate

Before enabling this function, you must first have at least one Ad-Hoc recipient in the program:

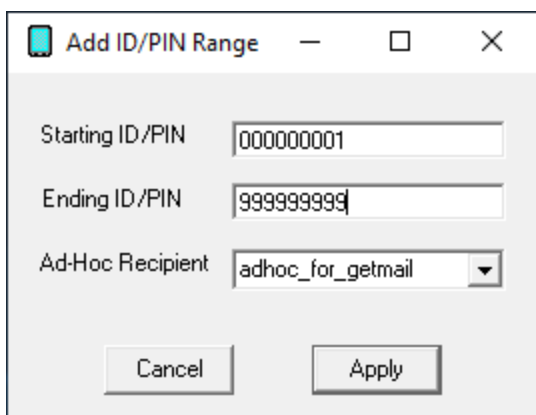
- 1) [Create an Ad-Hoc recipient](#).
- 2) Under the 'Enabled Services' section of the recipient, make sure GetMail is checked.
- 3) Select the carrier that messages should be sent to via the Ad-Hoc function.

Once the Recipient has been configured, check "Enable Ad-Hoc Mail".



First ID	Last ID	Ad-Hoc Recipient
----------	---------	------------------

- 1) Click Add.



Starting ID/PIN: 000000001

Ending ID/PIN: 999999999

Ad-Hoc Recipient: adhoc_for_getmail

- 2) Enter the Starting ID/PIN.
- 3) Enter the Ending ID/PIN.
- 4) Select an Ad-Hoc Recipient from the drop-down menu
- 5) Click Apply.

PageGate Admin: How to Configure PageGate

6

Method 2

To configure the GetMail module to accept email with the ID/PIN numbers in the subject line of incoming email:

- 1) [Create an Ad-Hoc recipient](#).
- 2) Under the 'Enabled Services' section of the recipient, make sure GetMail is checked.
- 3) Select the carrier that messages should be sent to via the Ad-Hoc function.

To use this new ad-hoc email, send email addressed to the ad-hoc recipient (recipientname@getmaildomain.com) and use one or more ID/PIN numbers (separated by spaces or commas) in the subject line.

PageGate will then use the carrier associated with this ad-hoc recipient to deliver a copy of the email message to each ID/PIN number in the subject.

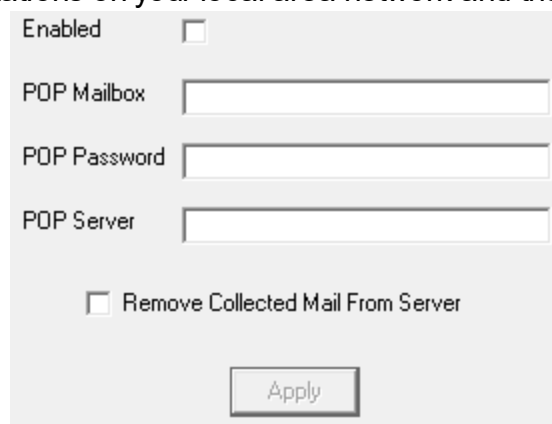
Domain POP

PageGate can collect all email for an entire domain from a single mailbox. Most Internet Service Providers have the ability to put all email addressed to your domain into a single POP mailbox. This is commonly known as 'Domain POP Mail' or 'Mailbagging'.

It works well with Dial-Up Internet connections or Internet connections that don't have a static IP address.

PageGate will try to collect mail from the mailbox according to the [POP Interval](#). Once the messages have been retrieved, PageGate will sort them into local mailboxes that must set up.

This setup can be used to give the appearance of having a dedicated, in-house mail server to both the workstations on your local area network and the Internet.



Enabled

POP Mailbox

POP Password

POP Server

Remove Collected Mail From Server

Apply

To configure the Domain POP section of PageGate:

PageGate

- 1) Obtain a registered domain name. This can generally be done by contacting your Internet Service Provider or a Domain Hosting Site.
- 2) Have the host of your registered domain name set up mailbagging (domain pop mail)
- 3) If you will be using Dial-Up Networking to connect to the Internet, see the [DUN section of the manual](#).
- 4) Check Enabled in the Domain POP section of the GetMail Settings.
- 5) Enter the name of the mailbox your ISP has configured. Note: Do not include the @domain.com part in the 'POP Mailbox' field.
- 6) Enter the password to access the mailbox in the POP Password field.
- 7) Enter the name of the mail server in the POP Server field. Note: The POP Server is the portion of the address after the @ symbol. Do not include the @ symbol.
- 8) If you want PageGate to remove collected messages from the Domain POP Email Account, put a check in "Remove Collected Mail From Server".
- 9) Click Apply.
- 10) Set up one or more recipients in PageGate. By default, the recipient's name will be the name of their email address.

DUN

DUN is an acronym for Dial-Up Networking.

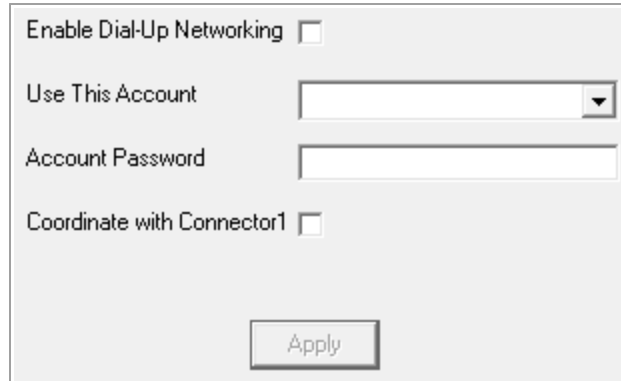
PageGate can use the Dial-Up Networking function of Windows to connect to the Internet for collecting email. A Dial-Up Networking Connection must first be setup within the operating system in order for this to work.

When the DUN portion of the [GetMail Settings](#) is enabled, the POP Interval setting will be used to determine how often PageGate will connect to the Internet to collect email. After PageGate is done receiving messages from the email account(s), it will automatically disconnect from the Internet and wait for the next POP Interval.

If the Dial-Up Networking account is already connected when the POP Interval happens, PageGate will use the current connection but will not disconnect when finished. Basically, PageGate assumes that another application is controlling the Dial-Up session and will not interfere with it.

PageGate Admin: How to Configure PageGate

6



Setting	Function
Enable Dial-Up Networking	This option enables and disables the Dial-Up Network portion of PageGate's GetMail module.
Use This Account	An existing Dial-Up networking connection must be configured in Windows before this setting can be used. Select a Dial-Up networking connection from the drop-down list.
Account Password	Enter the password required by the selected Dial-Up Networking connection.
Coordinate with Connector1	If you don't have access to multiple modems and require the primary Connector to function in tandem with the Dial-Up Networking portion of PageGate, place a check here.

Status

This section displays the current status of the GetMail module.



If the status of GetMail is stopped, it indicates that the module is not running and this can be caused by a few things.

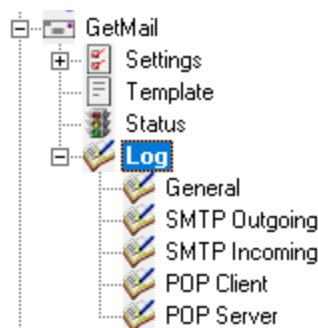
- GetMail will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- Enabled must be checked in the [settings of the GetMail module](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running

PageGate

PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

Log

This section contains a record of all functions performed by GetMail and the PageGate Mail Server service. As the GetMail module has many functions, it has five separate logs that correspond to it.



- [General](#)

This log is a record of information and traffic processed by the GetMail module.

- [SMTP Outgoing](#)

This log is a record of all outgoing SMTP transactions negotiated by the PageGate Mail Server.

- [SMTP Incoming](#)

This log is a record of all incoming SMTP transactions negotiated by the PageGate Mail Server.

- [POP Client](#)

This log is a record of all POP mail collection sessions negotiated by the PageGate Mail Server.

- [POP Server](#)

This log is a record of all incoming POP mail collection requests negotiated by the PageGate Mail Server.

PageGate

General

This log is a record of information and traffic processed by the GetMail module.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

SMTP Outgoing

This log is a record of all outgoing SMTP transactions negotiated by the PageGate Mail Server.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

SMTP Incoming

This log is a record of all incoming SMTP transactions negotiated by the PageGate Mail Server.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

POP Client

This log is a record of all POP mail collection sessions negotiated by the PageGate Mail Server.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

POP Server

This log is a record of all incoming POP mail collection requests negotiated by the PageGate Mail Server.

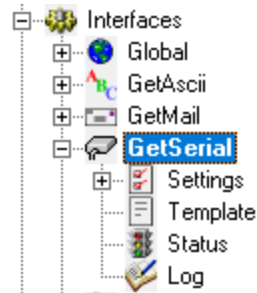


Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

GetSerial

This interface allows PageGate to monitor a serial/RS232 port (virtual or physical) for raw data ascii strings to be parsed in to outbound messages. When a data string is sent to a monitored serial port, GetSerial pattern matches sections of the data string to parse information for messaging.

PageGate Admin: How to Configure PageGate



- [Settings](#)

This section controls the implementation of the GetSerial API.

- [Status](#)

This section displays the running/not running status of the interface.

- [Template](#)

This section allows you to configure a custom template for messages processed by this API.

- [Log](#)

This section is a record of all activity performed by the interface.

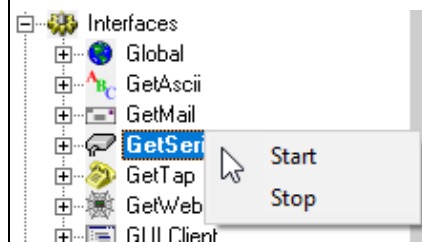
Right clicking on GetSerial will provide a context menu with two options.

- [Start](#)

Selecting this option will start the GetSerial interface, if stopped.

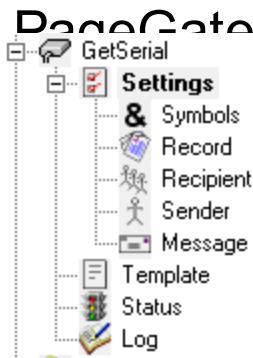
- [Stop](#)

Selecting this option will stop the GetSerial interface, if started.



Setting	Function
Enabled	When the Enabled checkbox is checked, it tells the PageGate server to run the GetSerial module.
Serial Port	This field determines the Communications (COM) Port the GetSerial module will monitor for incoming data.
Baud Rate	This setting must match the Baud Rate the sending device uses.
Parity	This setting must match the Parity the sending device uses.
Data Bits	This setting must match the Data Bits the sending device uses.
Stop Bits	This setting must match the Stop Bits the sending device uses.
Handshaking	If the sending device uses handshaking for flow control, GetSerial must mirror the handshaking method (Xon/Xoff or RTS). If the sending device does not use handshaking, this field must be set to None.
Init String	Init strings are not typically used with Serial Connections but may be used if a modem is attached to the same COM port specified in the Serial Port setting.

Expanding the Settings section will provide the following menu items:



This is where you'll configure PageGate to recognize the patterns in the stream of data being read in on the RS232/Serial line.

- [Symbols](#)

This section determines certain special characters and their functions.

- [Record](#)

This section is where you configure the pattern recognition to determine what represents the start and end of valid data to process.

- [Recipient](#)

This section is where you configure the pattern recognition to determine what represents who the messages should go to. This section can also be set to a static recipient, to indicate that all traffic processed from the serial line should be sent to the specified individual or group.

- [Sender](#)

This section is where you configure the pattern recognition to determine what represents who or what sent the messages. This section can also be set to a static sender to indicate that all traffic processed from the serial line should declare that it is from the specified value.

- [Message](#)

This section is where you configure the pattern recognition to determine what represents the body of the message to be delivered. This section can also be set to a static message to provide the message text to be delivered.

PageGate Admin: How to Configure PageGate

6

This section determines certain special characters and their functions.

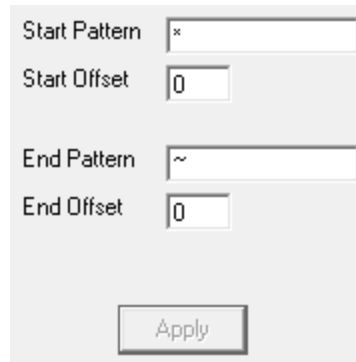
Single Character Wildcard	<input type="text" value="?"/>
Multiple Character Wildcard	<input type="text" value="*"/>
End Of Record Symbol	<input type="text" value="~"/>
Ascii Value Delimiter	<input type="text" value="\"/>

PageGate

Setting	Function
Single Character Wildcard	<p>When searching for pattern matches in the data strings on the RS232/Serial line, this character can be used to stand in for any single character.</p> <p>For example, let's say you use Fire?</p> <p>That would tell the GetSerial module to recognize any word that began with Fire and had one extra letter on the end. So, it would recognize Fires or Fired.</p> <p>As another example, let's say you use ??/??/????</p> <p>That would tell GetSerial to recognize any date stamp in the two digit month, two digit day, four digit year configuration.</p>
Multiple Character Wildcard	<p>When searching for pattern matches in the data strings on the RS232/Serial line, this character can be used to stand in for any character or group of characters.</p> <p>For example, let's say you use Fire*</p> <p>That would tell the GetSerial module to recognize any instance of the word Fire, regardless of whether there are any characters after the word. So, it would recognize Fire or Fired or Fires but it would also recognize the Fire in Fire Sale or Fire Department.</p> <p>As another example, let's say you use *age*</p> <p>That would tell GetSerial to recognize any reference to the three letters age close to each other, regardless of whether that was in the word PageGate or NotePage or ageless or damage. Any instance of the three characters together will be recognized.</p>
End of Record Symbol	<p>This character stands for the end of record symbol.</p>
ASCII Value Delimiter	<p>This symbol is used to help represent non-printable characters that normally can't be typed like horizontal tabs, carriage returns and line feeds. This character is always followed three numbers which represent the decimal value for a character.</p> <p>For example, this represents a carriage-return: \013</p> <p>For a full list of ascii codes, visit http://www.asciitable.com/</p>
Apply	<p>Applies all changes made to the Symbols.</p>

Record

This section is where you configure the pattern recognition to determine what represents the start and end of a message sequence.



The screenshot shows a configuration window with four input fields and one button. The fields are labeled 'Start Pattern', 'Start Offset', 'End Pattern', and 'End Offset'. The 'Start Pattern' field contains the character '*'. The 'Start Offset' field contains the number '0'. The 'End Pattern' field contains the character '~'. The 'End Offset' field contains the number '0'. Below these fields is an 'Apply' button.

- **Start Pattern**

This value determines the start of relevant information. For example, let's say this is a data string read in on the RS232/Serial line:

```
CODE 17 START A04 Refrigeration Unit 003 Temperature High 84.31 N773 END  
A9.3178
```

You would want to use START as the Start Pattern as that begins the section of information relevant for your message. Start patterns are included in the record processed unless you specify the otherwise with the Start Offset.

- **Start Offset**

This value determines how many characters before or after the identified start pattern to begin character processing. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will begin the data record 5 characters after the first character in the Start Pattern. A value of -5 means that GetSerial will begin the data record 5 characters before the first character the Start Pattern. To use our earlier example, if we don't want the word START to be a part of the record processed, we would set the Start Offset value to 5. That way, the data record begins five characters after the first character in START.

- **End Pattern**

This value determines the end of relevant information. For example, let's say this is a data string read in on the RS232/Serial line:

```
CODE 17 START A04 Refrigeration Unit 003 Temperature High 84.31 N773 END  
A9.3178
```

You would want to use END as the End Pattern as that ends the section of information relevant for your message. End patterns are included in the record processed unless you specify the otherwise with the End Offset.

- **End Offset**

PageGate Admin: How to Configure

PageGate

This value determines where in the data stream the record of relevant information ends and does so by determining how many characters before or after the identified End Pattern to begin. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will end the data record 5 characters after the last character in the End Pattern. A value of -5 means that GetSerial will end the data record 5 characters after the last character the End Pattern. To use our earlier example, if we don't want the word END to be a part of the record processed, we would set the Start Offset value to -3. That way, the data record ends three characters before the last character of END.

Recipient

This section is where you configure the pattern recognition to determine what represents who the messages should go to.

The screenshot shows a configuration panel with two main sections: 'Static Recipient' and 'Dynamic Recipient'. The 'Dynamic Recipient' section is selected with a radio button. It includes input fields for 'Start Pattern' (containing '*'), 'Start Offset' (containing '0'), 'End Pattern' (containing '~'), and 'End Offset' (containing '0'). There is a checkbox for 'Use Lookup Table' which is unchecked, and an 'Edit Table' button below it. At the bottom of the panel is an 'Apply' button.

There are two possible configurations you can use:

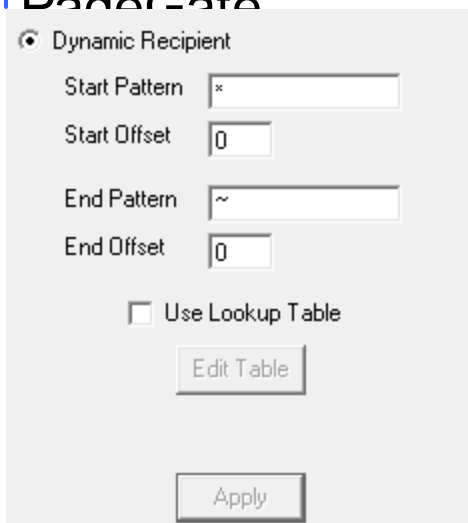
- Static Recipient

This screenshot shows the 'Static Recipient' configuration option selected with a radio button. It features a single input field labeled 'Recipient' with a dropdown arrow on the right side.

Selecting this option tells the GetSerial module that all messages processed from the RS232/Serial line should go to the specified recipient or group.

- Dynamic Recipient

PageGate



Dynamic Recipient

Start Pattern

Start Offset

End Pattern

End Offset

Use Lookup Table

Selecting this option tells the GetSerial module to look for specific patterns that denote where the recipient information is located in within the [record](#).

• **Start Pattern**

This value determines the first character in the recipient value.

For example, let's say this is a data string identified as the [record](#) and let's also say that the first three characters represent the recipient or group this message should go to:

A04 Refrigeration Unit 003 Temperature High 84.31 N773

or another example:

B16 Refrigeration Unit 007 Temperature High 92.74 N712

or another example:

A29 Refrigeration Unit 005 Temperature High 85.93 N798

You would want to use * as the Start Pattern as the recipient information is the first value in the string of data. Start patterns are included in the recipient processed unless you specify the otherwise with the Start Offset.

• **Start Offset**

This value determines how many characters before or after the identified start pattern to accept as the recipient value. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will begin looking for the recipient value 5 characters after the first character in the Start Pattern. A value of -5 means that GetSerial will begin looking for the recipient value 5 characters before the first character the Start Pattern.

To use our earlier example, we would want a Start Offset of 0 because we want everything from the first character in the data stream.

PageGate Admin: How to Configure PageGate

- **End Pattern**

This value determines the last character in the recipient value.

Using our earlier example, we've already specified that the recipient value starts at the very first character in the stream.

A04 Refrigeration Unit 003 Temperature High 84.31 N773

or another example:

B16 Refrigeration Unit 007 Temperature High 92.74 N712

or another example:

A29 Refrigeration Unit 005 Temperature High 85.93 N798

Assuming ? represents the [single character wildcard value assigned in the Symbols section](#), you would want to use ??? as the End Pattern as the first three character sequence in the data represents the recipient value.

- **End Offset**

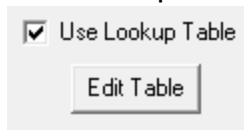
This value determines how many characters before or after the identified End Pattern to stop character recognition. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will stop looking for the recipient value 3 characters after the first character in the End Pattern. A value of -5 means that GetSerial will stop looking for the recipient value 5 characters before the first character the End Pattern.

To use our earlier example, since we have ??? specified as the End Pattern and that three character value represents the recipient value, we would set the Start Offset value to 3. That way, the data record ends three characters after the first character in the file.

- **Lookup Table**

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed. To enable this feature, check Use Lookup Table and click Edit Table.

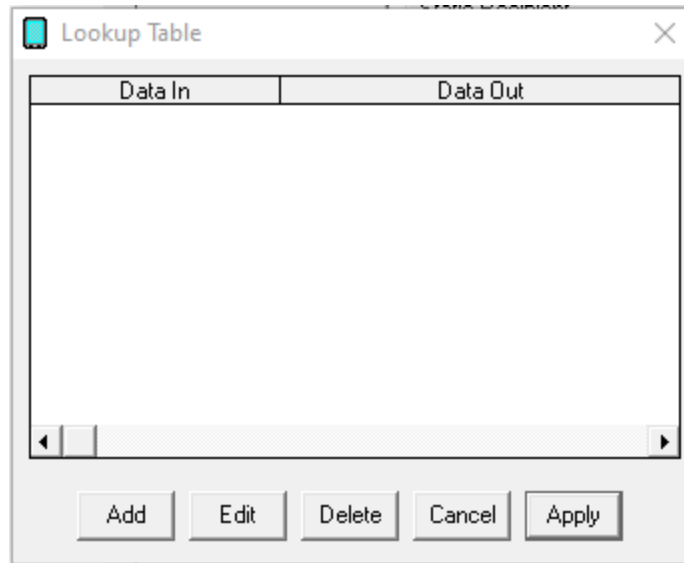


Lookup Table

This feature enables PageGate to perform basic search and replace commands on values read in as the Recipient, which allows you to interchange or translate values as needed.

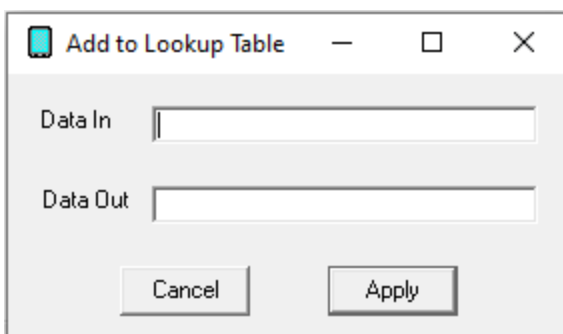
PageGate Admin: How to Configure PageGate

6



Button	Function
Add	Adds a new search and replace value.
Edit	Edits an existing search and replace value.
Delete	Removes an existing search and replace value.
Cancel	Closes the table without saving changes.
Apply	Applies updates made to the table.

When adding new entries to the table, you'll be presented with this dialog:



- **Data In**

This value represents what you want to search for. Wildcard characters can be used.

- **Data Out**

This value represents what you want to replace the text with.

For example, if we had a 'Data In' value of oranges and a 'Data Out' value of bananas, the program would replace any instance of the word oranges with the word bananas.

PageGate Admin: How to Configure PageGate

To provide a more practical example, let's use the [example from the GetSerial - Settings - Recipient section](#) of the documentation and say that we needed to translate A04 in to recipient or group name as you've entered it in to the PageGate Admin. You could use a 'Data In' value of A04 and a 'Data Out' value of ITDept so any messages that came in for A04 were routed to the group called ITDept in PageGate.

Sender

This section is where you configure the pattern recognition to determine what value represents who or what sent the messages read in from the RS232/Serial line.

Static Sender
 Sender

Dynamic Sender
 Start Pattern
 Start Offset
 End Pattern
 End Offset

Use Lookup Table

There are two possible configurations you can use:

- Static Sender

Static Sender
 Sender

Selecting this option tells the GetSerial module that all messages processed should identify themselves as having been sent from the specified value. This is the value that is used in the majority of GetSerial implementations.

- Dynamic Sender

The screenshot shows the 'Dynamic Sender' configuration panel. It includes four input fields: 'Start Pattern' with the value 'x', 'Start Offset' with the value '0', 'End Pattern' with the value '~', and 'End Offset' with the value '0'. Below these fields is a checkbox labeled 'Use Lookup Table' which is currently unchecked. Underneath the checkbox is an 'Edit Table' button. At the bottom of the panel is an 'Apply' button.

Selecting this option tells the GetSerial module to look for specific patterns that denote where the sender information is located in within the [record](#).

- **Start Pattern**

This value determines the first character in the sender value.

- **Start Offset**

This value determines how many characters before or after the identified start pattern to accept as the sender value. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will begin looking for the sender value 5 characters after the first character in the Start Pattern. A value of -5 means that GetSerial will begin looking for the sender value 5 characters before the first character the Start Pattern.

- **End Pattern**

This value determines the last character in the sender value.

- **End Offset**

This value determines how many characters before or after the identified End Pattern to stop character recognition. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will stop looking for the sender value 5 characters after the first character in the End Pattern. A value of -5 means that GetSerial will stop looking for the sender value 5 characters before the first character the End Pattern.

- **Lookup Table**

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed. To enable this feature, check Use Lookup Table and click Edit Table.

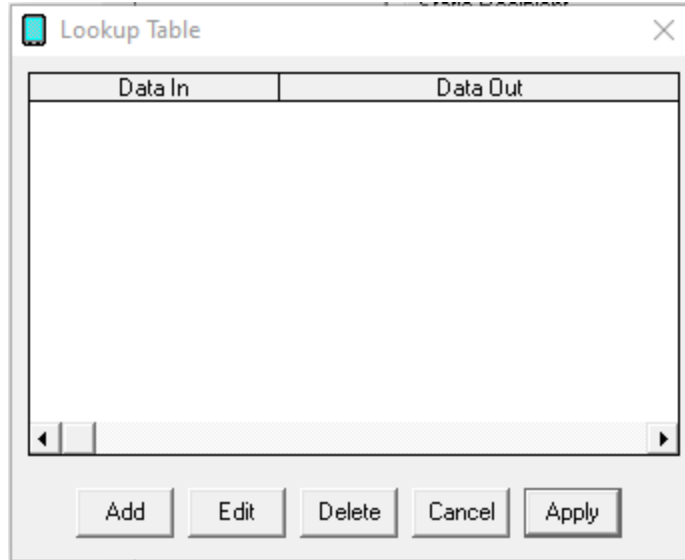
PageGate Admin: How to Configure

6



Lookup Table

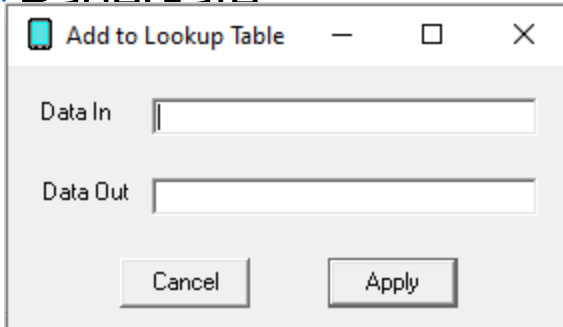
This feature enables PageGate to perform basic search and replace commands on values read in as the Sender, which allows you to interchange or translate values as needed.



Button	Function
Add	Adds a new search and replace value.
Edit	Edits an existing search and replace value.
Delete	Removes an existing search and replace value.
Cancel	Closes the table without saving changes.
Apply	Applies updates made to the table.

When adding new entries to the table, you'll be presented with this dialog:

PageGate Admin: How to Configure PageGate

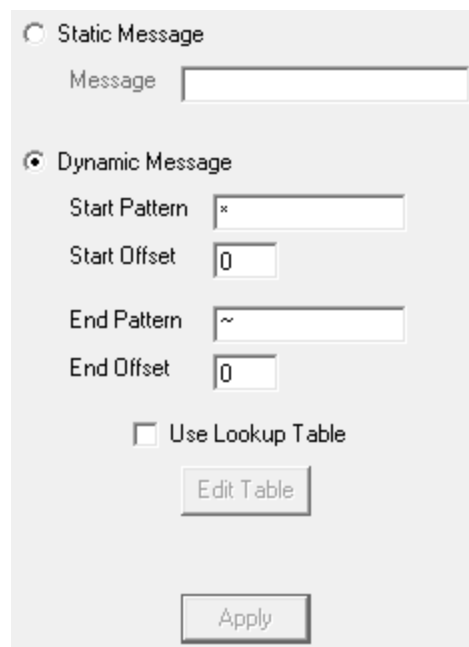


- **Data In**
This value represents what you want to search for. Wildcard characters can be used.
- **Data Out**
This value represents what you want to replace the text with.

For example, if we had a 'Data In' value of oranges and a 'Data Out' value of bananas, the program would replace any instance of the word oranges with the word bananas.

To provide a more practical example, let's say we wanted to translate A834 in to Dispatch Center. You would use a 'Data In' value of A834 and a 'Data Out' value of Dispatch Center.
Message

This section is where you configure the pattern recognition to determine what value represents the messages read in from the RS232/Serial line.



There are two possible configurations you can use:

- **Static Message**

PageGate

Static Message

Message

Selecting this option allows you to specify a pre-canned message to be delivered when activity is detected on the serial line.

- **Dynamic Message**

Dynamic Message

Start Pattern

Start Offset

End Pattern

End Offset

Use Lookup Table

Selecting this option tells the GetSerial module to look for specific patterns that denote where the message is located in within the [record](#).

- **Start Pattern**

This value determines the first character in the message value.

To use [the example we've been working with](#), the record originally started as:
A04 Refrigeration Unit 003 Temperature High 84.31 N773

We then configured GetSerial to treat A04 as the recipient value, which removes it from the data string. So, by the time the Message processing happens, this is the remaining data:

Refrigeration Unit 003 Temperature High 84.31 N773

As a result, we would want a Start Pattern of * to indicate that the message begins at the first remaining character.

- **Start Offset**

This value determines how many characters before or after the identified start pattern to accept as the message. This field can be set to positive or negative values.

PageGate Admin: How to Configure PageGate

6

For instance, a value of 5 means that GetSerial will begin looking for the message 5 characters after the first character in the Start Pattern. A value of -5 means that GetSerial will begin looking for the message 5 characters before the first character the Start Pattern.

Using our example, you would want to set the Start Offset to 0.

- **End Pattern**

This value determines the last character in the message.

In our example, you would want to set the End Pattern to the [End of Record symbol](#) in the [GetSerial - Settings - Symbols](#) section. This tells PageGate that the last character of the data line is the termination point for the message. So, by configuring a * for the Start Pattern and a ~ for the End Pattern, we're telling PageGate that all of the remaining data is relevant and needs to be in the body of the message delivered.

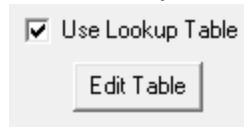
- **End Offset**

This value determines how many characters before or after the identified End Pattern to stop character recognition. This field can be set to positive or negative values.

For instance, a value of 5 means that GetSerial will stop looking for the message 5 characters after the first character in the End Pattern. A value of -5 means that GetSerial will stop looking for the message 5 characters before the first character the End Pattern.

- **Lookup Table**

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed. To enable this feature, check Use Lookup Table and click Edit Table.

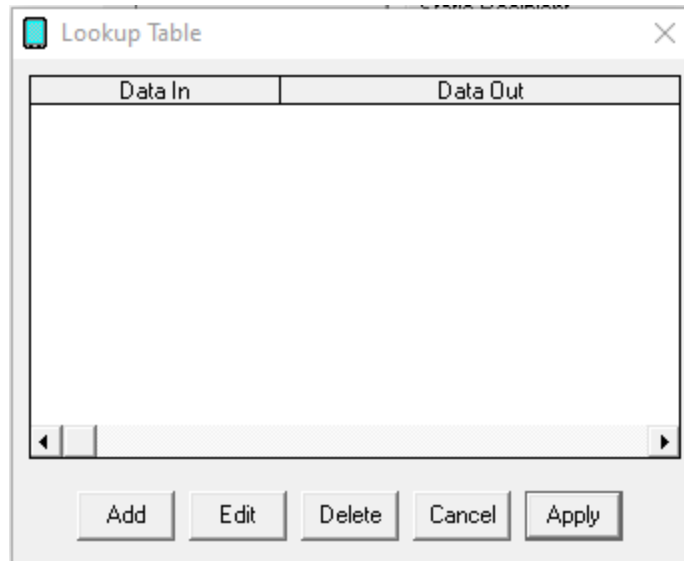


Lookup Table

This feature enables PageGate to perform basic search and replace commands on values in the body of the message, which allows you to interchange or translate values as needed.

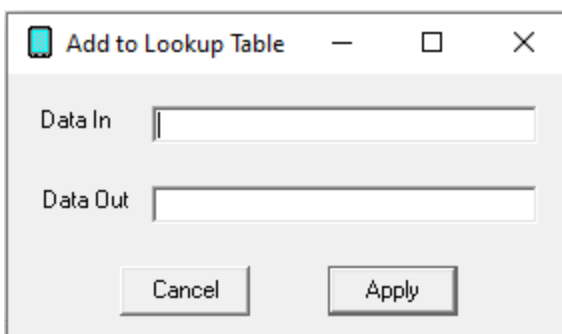
PageGate Admin: How to Configure PageGate

6



Button	Function
Add	Adds a new search and replace value.
Edit	Edits an existing search and replace value.
Delete	Removes an existing search and replace value.
Cancel	Closes the table without saving changes.
Apply	Applies updates made to the table.

When adding new entries to the table, you'll be presented with this dialog:



- **Data In**

This value represents what you want to search for. Wildcard characters can be used.

- **Data Out**

This value represents what you want to replace the text with.

For example, if we had a 'Data In' value of oranges and a 'Data Out' value of bananas, the program would replace any instance of the word oranges with the word bananas.

PageGate Admin: How to Configure PageGate

To provide a more practical example, let's use the [example from the GetSerial - Settings - Message section](#) of the documentation and say that we needed to translate 84.31 in to Building 14, Floor 3, Room 6. You would use a 'Data In' value of 84.31 and a 'Data Out' value of Building 14, Floor 3, Room 6. As another example, let's say you wanted to translate N773 in to Service contract Norwood, Account 773. You would use a 'Data In' value of N773 and a 'Data Out' value of Service contract Norwood, Account 773.

Status

This section displays the current status of the GetSerial module.



If the status of GetSerial is stopped, it indicates that the module is not running and this can be caused by a few things.

- GetSerial will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- Enabled must be checked in the [settings of the GetSerial module](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

PageGate Admin: How to Configure

Log

This log is a record of information and traffic processed by the GetSerial module.



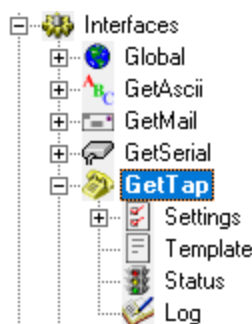
Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

GetTap

This interface allows PageGate to monitor a serial port or answer a modem call to receive TAP transmitted pages. This allows PageGate to function as a TAP receiver or a TAP terminal for any devices and systems that can transmit that protocol, such as a Zetron or

PageGate Admin: How to Configure PageGate

Motorola paging terminal as well as a few CAD systems. This also allows PageGate to function as a TAP receiver for any installation of [NotePager Pro](#), allowing satellite offices to submit messages for processing with a centralized, master server.



- [Settings](#)

This section controls the implementation of the GetTap API.

- [Status](#)

This section displays the running/not running status of the interface.

- [Template](#)

This section allows you to configure a custom template for messages processed by this API.

- [Log](#)

This section is a record of all activity performed by the interface.

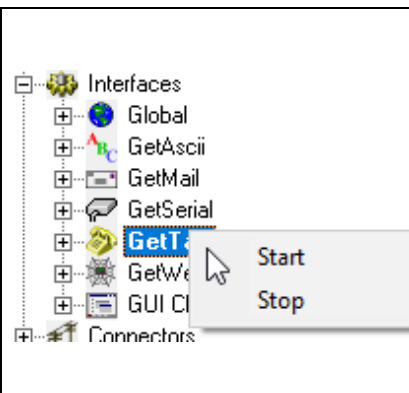
Right clicking on GetTap will provide a context menu with two options.

- [Start](#)

Selecting this option will start the GetTap interface, if stopped.

- [Stop](#)

Selecting this option will stop the GetTap interface, if started.



Settings



PageGate

Setting	Function
Enabled	When the Enabled checkbox is checked, it tells the PageGate server to run the GetTap module.
Serial Port	This field determines the Communications (COM) Port the GetTap module will monitor for incoming data. This field can be set to use the COM port of a modem or a serial port.
Baud Rate	Choose the baud rate at which incoming connections will be established. By default, the baud rate is 1200.
Parity	Choose the parity with which incoming connections will be established. By default, the parity is even.
Data Bits	Choose the data bits the incoming connections will be established with. By default, the data bits is 7.
Stop Bits	Choose the stop bits the incoming connections will be established with. By default, the stop bits is 1.
Init String	<p>The init string is a set of commands sent to the modem to configure it for incoming calls. The init string also turns certain modem connection features on and off. For example, most systems using the TAP protocol communicate at 1200 baud with no error correction or data compression. By default, the Init String field is set to ATSO=1 - this init string resets the modem to factory defaults and sets the modem to answer on the first ring.</p> <p>If you specified a Serial Port connected directly to a serial cable, instead of a modem, please set the Init String field to the word 'direct' (without the quotes).</p>

AdHoc

This function configures the GetTap module to accept ID/PIN values within a certain range and pass them through as the ID/PIN the message should ultimately be delivered to. For example, if the TAP system sending information in to the GetTap API uses ten digit phone number as the ID/PIN value, you could have that passed through in-state to the ad-hoc recipient without needing to configure a static recipient value to correspond to the ID passed.

First ID	Last ID	Ad-Hoc Recipient
----------	---------	------------------

Button	Function
Enable Ad-Hoc Tap	Enables the use of Ad-Hoc ranges for processing TAP ID's.
Add	Adds a new Ad-Hoc range.
Edit	Edits an existing Ad-Hoc range.
Delete	Removes an existing Ad-Hoc range
Apply	Applies updates made to the

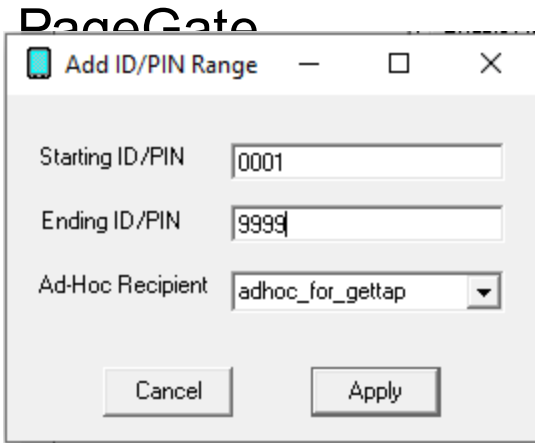
Before enabling this function, you must first have at least one Ad-Hoc recipient in the program:

- 1) Create a new [Ad-Hoc recipient](#).
- 2) Under the 'Enabled Services' section of the recipient, make sure GetTap is checked.
- 3) Select the carrier that messages should be sent to via the Ad-Hoc function.

Once the Recipient has been configured, check "Enable Ad-Hoc Tap".

Enable Ad-Hoc Map

- 1) Click Add.



- 2) Enter the Starting ID/PIN.
- 3) Enter the Ending ID/PIN.
- 4) Select an Ad-Hoc Recipient from the drop-down menu
- 5) Click Apply.

Status

This section displays the current status of the GetTap module.



If the status of GetTap is stopped, it indicates that the module is not running and this can be caused by a few things.

- GetTap will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- Enabled must be checked in the [settings of the GetTap module](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

PageGate Log

This log is a record of information and traffic processed by the GetTap module.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

GetWeb

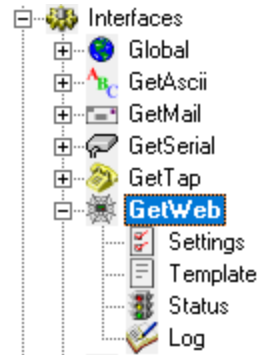
This interface provides a CGI executable that can be hosted on a web server and a series of pre-formatted templates to publish a website messaging interface. This allows you to host messaging websites on your local intranet or, if GetWeb is interfaced with an outward

PageGate Admin: How to Configure

6

PageGate

facing Web server, on the internet. Web Developers can also use this web server interface to send GET/POST requests to notify PageGate when a message needs to be sent, who the message should go to, who sent the message and what the message should say. This interface can also be used to enable PageGate to receive notifications from applications running on different platforms (Unix, Mac, etc.).



- [Settings](#)

This section controls the implementation of the GetWeb API.

- [Status](#)

This section displays the running/not running status of the interface.

- [Template](#)

This section allows you to configure a custom template for messages processed by this API.

- [Log](#)

This section is a record of all activity performed by the interface.

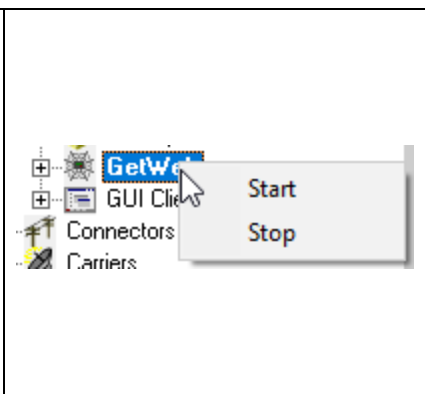
Right clicking on GetWeb will provide a context menu with two options.

- [Start](#)

Selecting this option will start the GetWeb interface, if stopped.

- [Stop](#)

Selecting this option will stop the GetWeb interface, if started.



The GetWeb module provides a CGI executable that can be implemented in one of two ways

Hosting messaging websites

PageGate Admin: How to Configure

PageGate

This option allows you to host a messaging website or series of messaging websites on your web server and GetWeb comes pre-loaded with a few stock templates to help you get started with this process. It should be noted that these templates are completely customizable and can be edited in any way you see fit.

- [GUI List Template Configuration](#)

This template should be tied to a group and provides a messaging website which offers a list of recipients and groups available to be messaged. Which recipients are groups are available to message is controlled by the member list of the group that the template is tied to. If a recipient or group is a member of the group the template is tied to, they show up on the website. If they aren't, they don't.

- [Adhoc Template Configuration](#)

This template should be tied to an Ad-Hoc recipient and provides a messaging website which offers a field to enter a phone number to message rather than a static list to choose from.

POST/GET request to GetWeb's CGI Executable

GetWeb's CGI executable can be used with either a POST or GET method.

The POST method can be used by creating a form on your webpage and pointing to the webgate.exe program in the FORM ACTION statement.

The GET method is more popular with scripting languages such as JavaScript and Java. It allows the GetWeb CGI to be called with a command similar to one that could be entered into a browser's URL field.

For more sophisticated webpages that use javascript, java, or another higher-end language than HTML, the webgate.exe CGI program can be called directly rather than using it only with an HTML form.

Standard Messaging

The three variables the webgate.exe CGI receiver is expecting to see are USER, FRM and MSG. USER specifies who the message should go to, FRM specifies who or what sent the message and MSG specifies the text to be sent.

Example:

```
http://www.mydomain.com/scripts/webgate.exe?  
USER=pageme&FRM=fromme&MSG=This+is+a+test
```

This would send a message to the recipient 'pageme', from 'fromme', with a message of 'This is a test'

Ad-Hoc Messaging

PageGate Admin: How to Configure

PageGate

The four variables the webgate.exe CGI receiver is expecting to see are USER, SUBJECT, FRM and MSG. USER specifies the ad-hoc recipient, SUBJECT specifies the phone numbers that should receive the message, FRM specifies who or what sent the message and MSG specifies the text to be sent.

Example:

`http://www.mydomain.com/scripts/webgate.exe?`

`USER=ADHOCMETHOD&Subject=PhoneNumber,PhoneNumber,PhoneNumber,etc&FRM=fromme&MSG=This+is+a+test`

This would send the message "This is a test" to the phone numbers specified in the "Subject" parameter.

Using Scripts

You can also find examples of Linux and Powershell scripts that can be used with GetWeb [in this section of the documentation](#).

Settings

This section is where you tell the GetWeb API what directory to reference on the web server to publish your messaging websites and what directory hosts the CGI executable.

Setting	Function
Enabled	When the Enabled checkbox is checked, it tells the PageGate server to run the GetWeb module.
Web Pages Path	This field determines the location on the web server where PageGate will publish your messaging website(s). This field should be set to a directory on the web server where new content should be created. If you have IIS installed locally and the Default Website referencing the default IIS location, set this field to: c:\inetpub\wwwroot\
CGI Data Path	This field determines the location on the web server where PageGate will host its CGI executable. This field should be set to the web server's scripting directory.

PageGate Admin: How to Configure PageGate

6

	For IIS installations on Windows 7, 8, 10, Server 2008, Server 2012 or Server 2016 please see the Configuring IIS 7.5, 8 and 10 section for steps on configuring IIS to allow GetWeb's CGI executable to function.
CGI URL	<p>This field determines the URL used in the FORM action and should be set to the URL and directory of the web server hosting webgate.exe</p> <p>If you have IIS installed locally and are using the default configuration described in the local IIS installation section of the documentation, set this to: <code>http://your-web-server/scripts/webgate.exe</code></p> <p>Be sure to replace your-web-server with the machine name or IP address of the PageGate server.</p>
Polling Interval	The setting in this field determines how often the GetWeb module looks for new messages in the CGI Data Path.

The GetWeb module provides a CGI executable that can be implemented in one of two ways

Hosting messaging websites

This option allows you to host a messaging website or series of messaging websites on your web server and GetWeb comes pre-loaded with a few stock templates to help you get started with this process. It should be noted that these templates are completely customizable and can be edited in any way you see fit.

- [GUI List Template Configuration](#)

This template should be tied to a group and provides a messaging website which offers a list of recipients and groups available to be messaged. Which recipients are groups are available to message is controlled by the member list of the group that the template is tied to. If a recipient or group is a member of the group the template is tied to, they show up on the website. If they aren't, they don't.

- [Adhoc Template Configuration](#)

This template should be tied to an Ad-Hoc recipient and provides a messaging website which offers a field to enter a phone number to message rather than a static list to choose from.

POST/GET request to GetWeb's CGI Executable

The webgate CGI can be used with either a POST or GET method.

The POST method can be used by creating a form on your webpage and pointing to the webgate.exe program in the FORM ACTION statement.

PageGate Admin: How to Configure

6

PageGate

The GET method is more popular with scripting languages such as JavaScript and Java. It allows the GetWeb CGI to be called with a command similar to one that could be entered into a browser's URL field.

For more sophisticated webpages that use javascript, java, or another higher-end language than HTML, the webgate.exe CGI program can be called directly rather than using it only with an HTML form. Here is a sample calling command that can be used in the URL field in a browser to call webgate.exe:

```
http://www.mydomain.com/scripts/webgate.exe?  
USER=pageme&FRM=fromme&MSG=This+is+a+test
```

This would send a message to the recipient 'pageme', from 'fromme', with a message of 'This is a test'

Just to note, the three variables the webgate.exe CGI receiver is expecting to see are USER, FRM and MSG. USER specifies who the message should go to, FRM specifies who or what sent the message and MSG specifies the text to be sent.

Configuring IIS 7.5, 8 and 10 for GetWeb

GetWeb can be configured to reference either a local installation of IIS or, if the appropriate directories on the web server are accessible on the network, a web server on the same network.

[Click here](#) for steps on how to configure a local installation of IIS for GetWeb.

[Click here](#) for steps on how to configure GetWeb to reference a web server accessible by UNC path.

Local IIS Installation

- 1) After installing IIS, if the directory does not already exist, create the directory c:\inetpub\scripts\
- 2) Open the PageGate Admin.
- 3) Go to Interfaces - GetWeb - Settings.
- 4) Set the "Web Pages Path" to: c:\inetpub\wwwroot\
- 5) Set the "CGI Data Path" to: c:\inetpub\scripts\
- 6) Set the "CGI URL" to: http://your-web-server/scripts/webgate.exe but replace your-web-server with the machine name or IP address of the PageGate server.
- 7) Check Enabled
- 8) Click Apply.
- 9) If prompted, you do want to enable this for all existing groups and recipients.

PageGate Admin: How to Configure PageGate

6

- 10) Open the IIS Admin and select the server object (the first thing in the list in the menu on the left side).
- 11) Open Handler Mappings.
- 12) Right click CGI-exe, left click Edit Feature Permissions
- 13) Check Execute.
- 14) Click OK.
- 15) Select the Server Object again.
- 16) Open ISAPI and CGI Restrictions.
- 17) Right click in the box and select Add.
- 18) Set the ISAPI or CGI path to: c:\inetpub\scripts\webgate.exe
- 19) Set the Description to: GetWeb CGI API
- 20) Check "Allow extension path to execute".
- 21) Click OK.
- 22) Right click Default Web Site, left click Add Virtual Directory.
- 23) Set the Alias to: scripts
- 24) Set the Physical path to: c:\inetpub\scripts\
- 25) Click OK.
- 26) Right click on the new scripts virtual directory, left click Edit Permissions.
- 27) Select the Security tab.
- 28) Click Edit.
- 29) Click Add.
- 30) Type Domain Users, left click Check Names.
- 31) Click OK.
- 32) Grant them the Modify permission.
- 33) Click Add.
- 34) Type IUSR, left click Check Names.
- 35) If prompted, select the local IUSR account.
- 36) Click OK.
- 37) Grant them the Modify permission.
- 38) Click Add.
- 39) Type Domain Admin, left click Check Names.
- 40) Click OK.

PageGate Admin: How to Configure

PageGate

41) Grant them the Full Control permission.

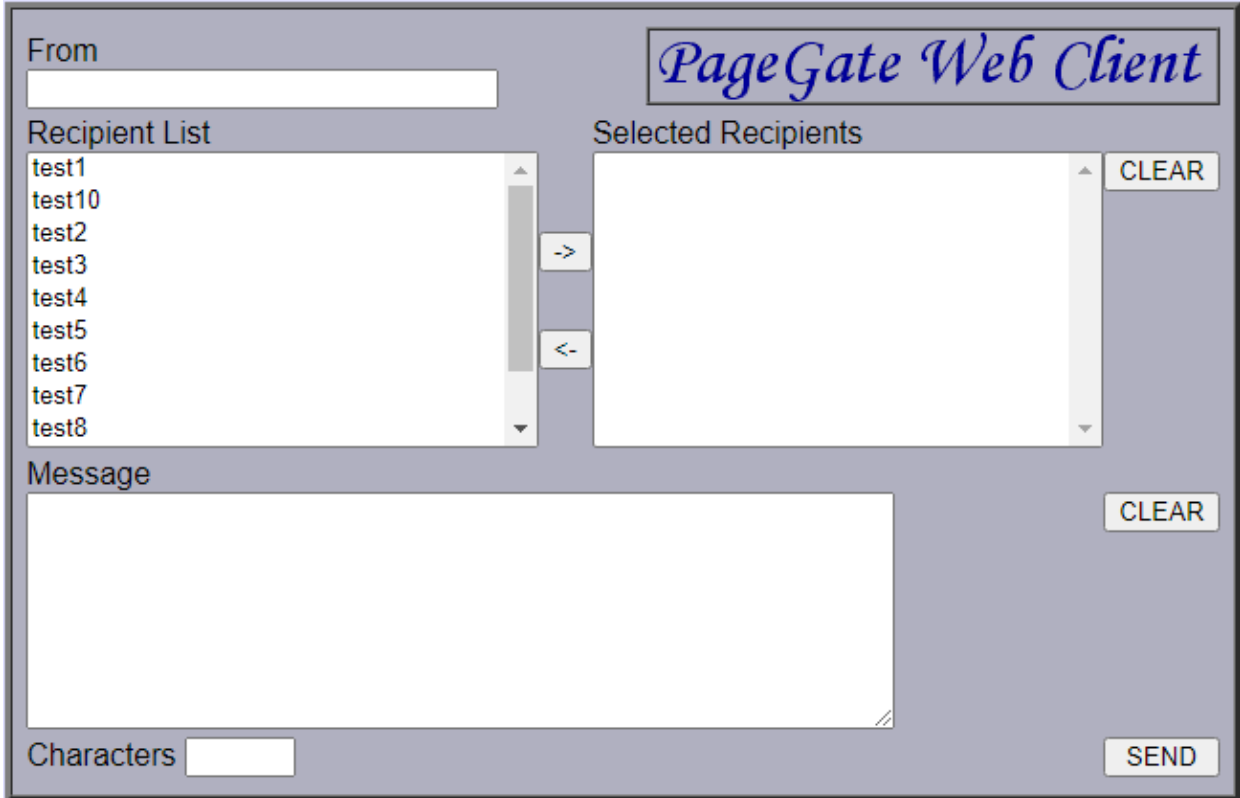
Network Web Server

- 1) Open the IIS Admin and select the server object.
- 2) Open Handler Mappings.
- 3) Right click CGI-exe, left click Edit Feature Permissions
- 4) Check Execute.
- 5) Click OK.
- 6) Select the Server object again.
- 7) Open ISAPI and CGI Restrictions.
- 8) Right click in the box and select Add.
- 9) Set the ISAPI or CGI path to the directory on the web server where webgate.exe is being hosted. This should correspond to the folder specified in GetWeb's [Settings](#).
- 10) Set the Description to: GetWeb CGI API
- 11) Check "Allow extension path to execute".
- 12) Click OK.

GUI List Template Configuration

NOTE: You must configure the [GetWeb API's settings](#) before you can implement this.

This template will format a messaging website to provide a list of recipients and groups available to be messaged and the page will look something close to the image shown below.



To configure GetWeb to publish this style of messaging website, you'll first need to create a [Group](#) to tie the webpage to as the member list of the group controls who is available in the Recipient List shown above. Once you have the group created, this is what you'll want to do:

- 1) Open the PageGate Admin.
- 2) Go to Groups - <YourGroupName> - WebPage.

PageGate Admin: How to Configure

PageGate

The screenshot shows the PageGate configuration form. At the top, there is a 'Template File' dropdown menu. Below it are ten text input fields labeled 'Text1' through 'Text10'. Underneath these are five image input fields labeled 'Image1' through 'Image5'. At the bottom, there are three checkboxes: 'Custom Page', 'Group List', and 'Full Names'. An 'Apply' button is located below the checkboxes.

3) In the 'Template File' dropdown menu, select: guilist.htm

4) In the Image1 field, enter: pagegatewebclient.gif

5) Check Group List.

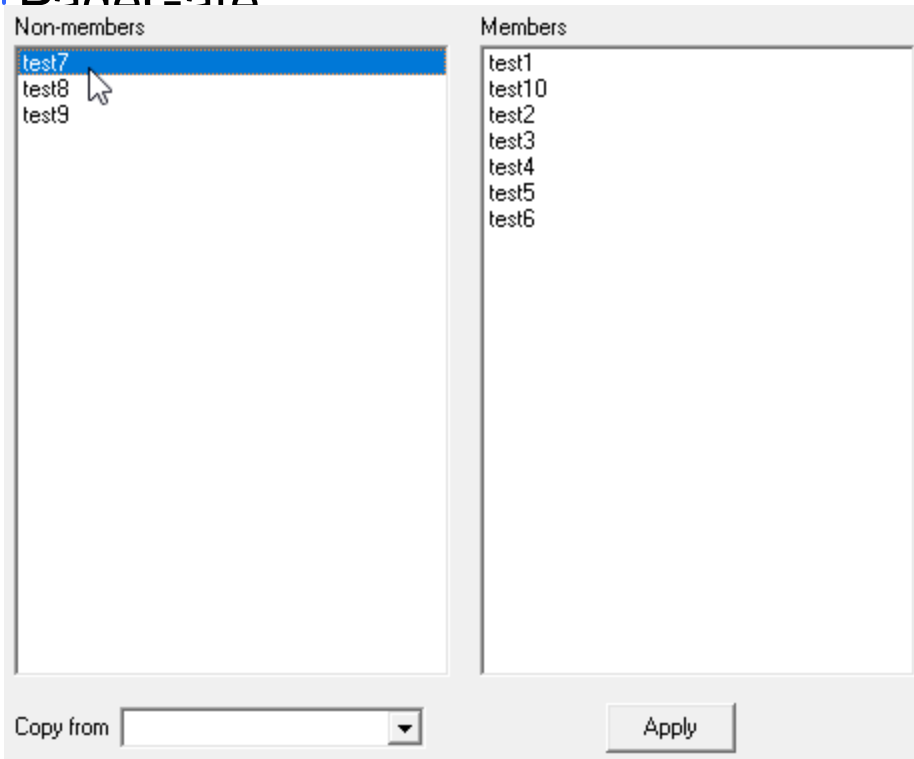
6) Check Full Names.

The screenshot shows the PageGate configuration form after several changes. The 'Template File' dropdown menu now displays 'guilist.htm'. The 'Group List' and 'Full Names' checkboxes are now checked. The 'Apply' button remains at the bottom.

7) Go to the Members sub-section of your new webpaging group.

8) Add all recipients and groups who should be available to message

PageGate Admin: How to Configure PageGate



9) Click Apply.

As soon as you do, PageGate will publish a directory to the [Web Pages path](#) specified in the [GetWeb Settings](#) that matches the name of the group and publish the index.htm and default.htm files to it. This makes a new messaging website, accessible by going to `http://<yourwebserver>/<yourgroupname>/`

For example, let's say that our web server's host name is webhost and that our group's name was webpaging. You would go to `http://webhost/webpaging/`

Adhoc Template Configuration

NOTE: You must configure the [GetWeb API's settings](#) before you can implement this.

This template should be tied to an [Ad-Hoc recipient](#) and provides a messaging website which offers a field to enter a phone number to message rather than a static list to choose from.

PageGate Admin: How to Configure PageGate

6

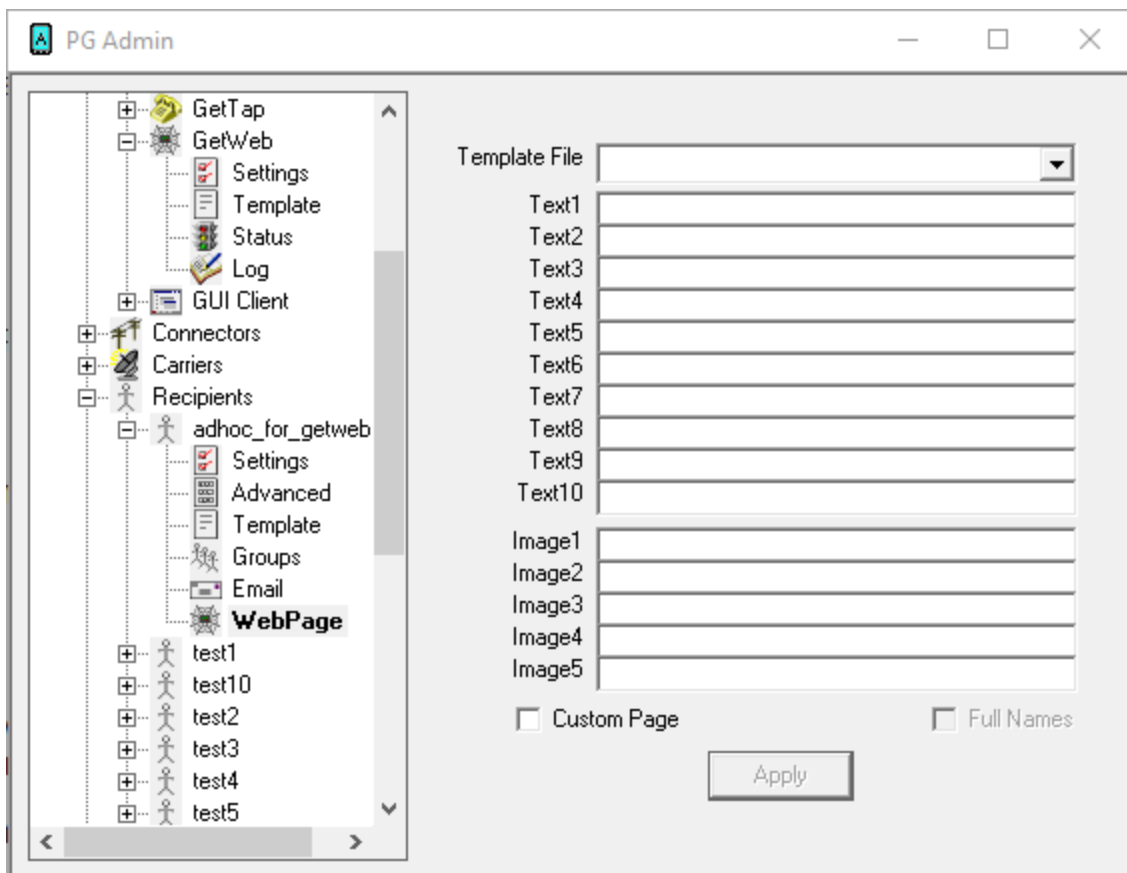
TO

FROM

MESSAGE

To configure GetWeb to publish this style of messaging website, you'll first need to create an [ad-hoc recipient](#) to tie the template to. Once you have the [ad-hoc recipient](#) created, this is what you'll want to do:

- 1) Open the PageGate Admin.
- 2) Go to Recipients - <YourAdHocRecipient> - WebPage.



- 3) In the 'Template File' dropdown menu, select: ad-hoc.htm

PageGate Admin: How to Configure PageGate

4) Click Apply.

As soon as you do, PageGate will publish a directory to the [Web Pages path](#) specified in the [GetWeb Settings](#) that matches the name of the ad-hoc recipient and publishes the index.htm and default.htm files to it. This makes a new messaging website, accessible by going to `http://<yourwebserver>/<youradhocrecipientname>/`

For example, let's say that our web server's host name is webhost and that our group's name was ad-hoc. You would go to `http://webhost/ad-hoc/`

Status

This section displays the current status of the GetWeb module.

If the status of GetWeb is stopped, it indicates that the module is not running and this can be caused by a few things.

- GetWeb will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- Enabled must be checked in the [settings of the GetWeb module](#).

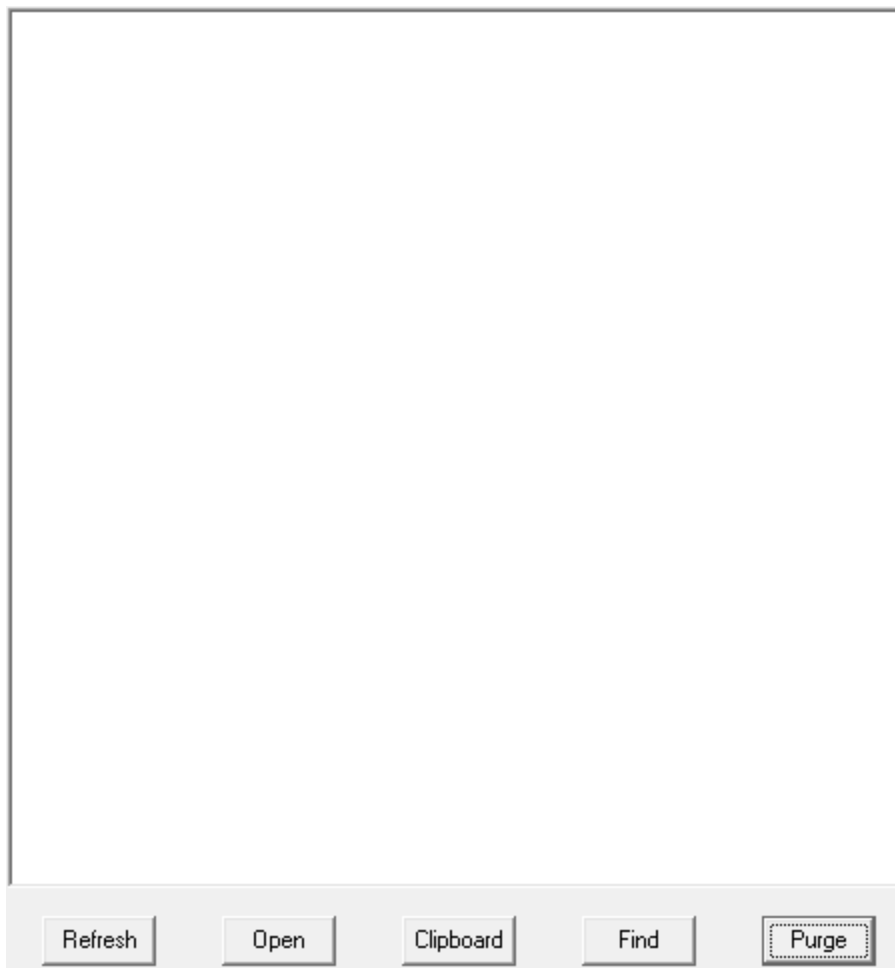
PageGate Admin: How to Configure

PageGate

- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

Log

This log is a record of information and traffic processed by the GetWeb module.



PageGate

Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

Supported Scripts

It's quite easy to use a script to pass information to GetWeb's CGI executable. Below, you'll find examples of both Linux and Powershell scripts to use as a guide:

Linux Script

```
#!/bin/bash
```

```
# set each part of the URL to send
```

```
UrlToCgi=$(printf "http://127.0.0.1/scripts/webgate.exe")
```

```
RecipientVar=$(printf "?USER=")
```

```
Recipient=$(printf "pageyou")
```

```
SenderVar=$(printf "&FRM=")
```

```
Sender=$(printf "fromme")
```

```
MessageVar=$(printf "&MSG=")
```

```
# set message to all the text sent on the command line
```

```
Message=$@
```

```
# url encode message text
```

```
Message=$(php -r "echo urlencode('$Message');")
```

```
# build the request URL
```

```
FullUrl=$UrlToCgi$RecipientVar$Recipient$SenderVar$Sender$MessageVar$Message
```

```
# send the request
```

```
wget -O- "$FullUrl"
```

Powershell Script

```
# set each part of the URL to send
```

```
$UrlToCgi="http://127.0.0.1/scripts/webgate.exe"
```

```
$RecipientVar="?USER="
```

```
$Recipient="pageyou"
```

PageGate Admin: How to Configure

PageGate

```

$SenderVar="&FRM="
$Sender="fromme"
$MessageVar="&MSG="
# set message to all the text sent on the command line
$Message=$args
# url encode message text
[uri]::EscapeUriString($Message)
# build the request URL
$FullUrl=$UrlToCgi + $RecipientVar + $Recipient + $SenderVar + $Sender +
$MessageVar + $Message
# send the request
Invoke-WebRequest -Uri $FullUrl

```

GUI Client

The GUI Client API is a Windows based program installed on workstations on the same network as the PageGate server. This program provides a graphical user interface to send and schedule messages. Since the PageGate Client is its own application, the PageGate Admin's GUI Client section is only used to enable or disable this interface, modify the template tied to this interface, to check the status of the interface and to review the log of traffic submitted by the workstation clients.



- [Settings](#)

This section controls whether the GUI Client interface is enabled or disabled.

- [Status](#)

This section displays the running/not running state of the interface.

- [Template](#)

This section allows you to configure a custom template for messages processed by this API.

- [Log](#)

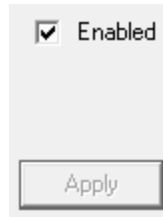
This section displays a record of all messages submitted by the GUI Client.

PageGate Admin: How to Configure PageGate

6

[Click here](#) for information on how to install, configure and implement the PageGate GUI Client.

Settings



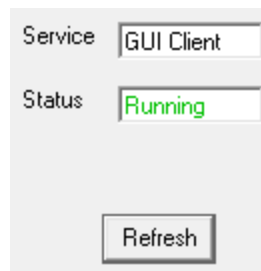
The only setting for the GUI Client contained within the PageGate Admin is whether the GUI Client module is Enabled or Disabled. By default, the GUI Client is set to Enabled.

To disable the GUI Client interface, uncheck Enabled and click Apply.

To enable the GUI Client interface, check Enabled and click Apply.

Status

The status subsection of GUI Client will show you if the GUI Client module is currently running or stopped.



If the Status of GUI Client is Stopped, that indicates that either the module is not enabled or the PageGate server is not active.

- Enabled must be checked in the [settings of the GUI Client module](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

Log

This log is a record of information and traffic submitted by workstations using the GUI Client.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

Additional APIs

- [SNPP Server](#)

This API allows you to host an SNPP server to receive messages in the SNPP protocol.

- [TCP Server Module](#)

This module allows PageGate to run a TCP server/listener, enabling it to accept TCP packets from any source.

PageGate Admin: How to Configure

PageGate

- [UDP Server Module](#)

This module allows PageGate to run a UDP server/listener, enabling it to accept UDP packets from any source.

- [TAP to ASCII](#)

This API allows you to monitor a COM port (physical or virtual) for either modem or serial messages in the TAP protocol. When a monitored modem or serial port passes a TAP message, the TAP to ASCII module records the information and submits it to the GetASCII API for processing.

- [TAP IP Server](#)

This API allows you to receive the TAP protocol by TCP/IP instead of the standard modem or serial/RS232 line.

- [Serial to ASCII](#)

This API allows you to monitor a serial/RS232 port (physical or virtual) for raw data ASCII strings to be parsed in to outbound messages. When a data string is sent to a monitored serial port, the Serial to ASCII module records the information and submits it to the GetASCII API for processing.

- [Filter Pack](#)

The filter pack is a series of scripts that allow you to modify any input from any API based on nearly any selection criteria of your choosing. These scripts also allow you to read any plaintext content generated by another application; whether that's another program's log files, XML or HTML content, delimited text or anything that can be read as plaintext.

SNPP Server

This interface allows PageGate to function as an SNPP server to receive this protocol. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#).

Step 1: Create a directory for the SNPP Server

1) In Windows, create a new directory for the SNPP Server. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called SNPP as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\SNPP\).

2) In Windows, browse in to the PageGate [program directory](#).

PageGate

3) Move the snppserver.exe, pininfo.txt and snpp.ini files to the folder you created in step 1.

Step 2: Configure the SNPP Server's settings

1) The first thing we'll need to do is modify the snpp.ini file. Open the file in notepad.

To give you a little more information, the snpp.ini file controls the behavior of the SNPP Server. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

2) Change the OutputFolder= value to the directory you created in step 1. By default, C:\PageGateData\SNPP\

3) Change the LogFolder= value to your [logging directory](#). If you are unsure of what this folder is, open the PageGate Admin and go to the [Program - Settings](#) section. By default, C:\PageGateData\Logs\

4) Save the .ini file and close notepad.

5) Open pininfo.txt in notepad.

This file tells the SNPP server what range of numbers can be passed to the server. The format of the entries defined in this text file are:

"Starting PIN", "Ending PIN", "recipient type"

If you would like ID numbers in a range to be recognized as the recipient's name (not using [ad-hoc](#), but using numeric recipient names in PageGate), use the text "PG_Recipient" as shown below:

"0001", "9999", "PG_Recipient"

If you would like ID numbers in a range to be passed through PageGate as intact ID's to another source in an [ad-hoc](#) range, use the name of a [Recipient in PageGate whose type is Ad-Hoc](#).

"0000001", "4999999", "adhoc1"

To have it accept all potential alphanumeric values, use the following statements:

"*", "*", "PG_Recipient"

"*", "*", "adhocrecipient"

6) After declaring the appropriate ID ranges, save and close the file.

Step 3: Run the SNPP Server

PageGate Admin: How to Configure PageGate

6

- 1) Open a command prompt with elevated permission.
- 2) Change directory to the SNPP Server's program directory. By default, c:\PageGateData\SNPP\

```
snppserver /install
```
- 3) Type the following and hit enter:

```
snppserver /install
```
- 4) Exit the command prompt.
- 5) Open the Services list in Windows.
- 6) Go in to the Properties of the PageGate SNPP Server service.
- 7) Change the Startup Type to Automatic and Start the Service.

Step 4: Configure the GetASCII API to process the SNPP Server's messages

- 1) Go to Interfaces - GetASCII - Settings.
- 2) Set the 'Polling Directory' to the output folder. By default, C:\PageGateData\SNPP\

```
snppserver /install
```
- 3) Check Enabled.
- 4) Click Apply.
- 5) Go to Program - Settings.
- 6) In the 'Run on this server' section, check GetASCII
- 7) Click Apply.
- 8) If prompted for credentials, leave them blank and click Apply again.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the SNPP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc and it is a Standard file type.

TCP Server Module

This interface allows PageGate to function as a TCP server to receive TCP packets from a variety of sources. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#).

Step 1: Create a directory for the TCP Server

PageGate

1) In Windows, create a new directory for the TCP Server. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called TCP as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\TCP\).

2) In Windows, browse in to the PageGate [program directory](#).

3) Move the pgtcpserver.exe and pgtcp.ini files to the folder you created in step 1.

Step 2: Configure the TCP Server's settings

1) The first thing we'll need to do is modify the pgtcp.ini file. Open the file in notepad.

To give you a little more information, this INI file controls the behavior of the TCP Server. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

2) Configure the port the TCP server should listen to. By default, the port is set to 5002.

3) Change the OutputFolder= value to the directory you created in step 1. By default, C:\PageGateData\TCP\

4) Change the LogFolder= value to your [logging directory](#). If you are unsure of what this folder is, open the PageGate Admin and go to the [Program - Settings](#) section. By default, C:\PageGateData\Logs\

5) Next, we'll need to set the end of record delimiter but let's go in to what that means first. When the TCP Server is receiving information, it doesn't really know what in the data is important; it's just receiving TCP packets of data. The end of record delimiter tells the TCP server that the expressed series of characters denotes a break in the data stream.

For example, the TCP server could receive a series of 5 packets:

- a) This is
- b) a
- c) test message #
- d) This is a new
- e) message

If you set EndOfRecordDelimiter = #, that tells the TCP server that everything before every # received is a data block to be parsed.

PageGate

Put another way, the end of record delimiter tells the TCP server how to break down the information it's receiving in to processable chunks.

It's the "over" of a CB radio conversation.

5) Below that, there are RequiredText and RequiredTextOperator values that can be set.

RequiredText specifies text that must appear for a data block to be processed and you may specify multiple values by separating each with a | (pipe).

The RequiredTextOperator specifies whether you're using OR or AND logic.

For example, let's say you specify the following:

```
RequiredText=alarm|notification
```

```
RequiredTextOperator=AND
```

That tells the TCP server that both the words alarm AND notification need to appear in a received data block for that data block to be treated as a message to send.

As another example:

```
RequiredText=alarm|alert
```

```
RequiredTextOperator=OR
```

That tells the TCP server that either the words alarm or alert need to appear in a received data block for that data block to be treated as a message to send.

6) Next we have EndOfRecordTimeout, whose value is expressed in milliseconds. This value tells the TCP server how long to wait before timeout after receiving data, to accept a data record.

By default, this is set to 3000 milliseconds or 3 seconds.

7) RejectOnTimeout determines whether timeouts should post a rejection notice or not. By default, this is set to true.

8) AcceptResponse determines what the TCP server will echo to the sending system upon acceptance of a data block.

9) RejectResponse determines what the TCP server will echo to the sending system upon rejection of a data block.

10) Last, we have two optional settings: StaticRecipient and StaticSender

StaticRecipient allows you to specify the name of a specific recipient or group in PageGate that ALL traffic from the TCP server should be passed through.

PageGate Admin: How to Configure PageGate

6

StaticSender allows you to hard code a sender's name that should be associated with ALL traffic from the TCP server.

11) Save the .ini file and close notepad.

Step 3: Run the TCP Server

- 1) Open a command prompt with elevated permission.
- 2) Change directory to the TCP Server's program directory. By default, c:\PageGateData\TCP\
`cd c:\PageGateData\TCP\`
- 3) Type the following and hit enter:
`pgtcpserver /install`
- 4) Exit the command prompt.
- 5) Open the Services list in Windows.
- 6) Go in to the Properties of the PageGate TCP Server service.
- 7) Change the Startup Type to Automatic and Start the Service.

Step 4: Configure the GetASCII API to process the TCP Server's messages

- 1) Go to Interfaces - GetASCII - Settings.
- 2) Set the 'Polling Directory' to the output folder. By default, C:\PageGateData\TCP\
`C:\PageGateData\TCP\`
- 3) Check Enabled.
- 4) Click Apply.
- 5) Go to Program - Settings.
- 6) In the 'Run on this server' section, check GetASCII
- 7) Click Apply.
- 8) If prompted for credentials, leave them blank and click Apply again.

PageGate Admin: How to Configure PageGate

6

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the TCP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc and it is a Standard file type.

UDP Server Module

This interface allows PageGate to function as a UDP server to receive UDP packets from a variety of sources. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#).

Step 1: Create a directory for the UDP Server

- 1) In Windows, create a new directory for the UDP Server. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called UDP as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\UDP\).
- 2) In Windows, browse in to the PageGate [program directory](#).
- 3) Move the pgudpserver.exe and pgudp.ini files to the folder you created in step 1.

Step 2: Configure the UDP Server's settings

- 1) The first thing we'll need to do is modify the pgudp.ini file. Open the file in notepad.

To give you a little more information, this INI file controls the behavior of the UDP Server. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

- 2) Configure the port the UDP server should listen to. By default, the port is set to 5002.
- 3) Change the OutputFolder= value to the directory you created in step 1. By default, C:\PageGateData\UDP\
- 4) Change the LogFolder= value to your [logging directory](#). If you are unsure of what this folder is, open the PageGate Admin and go to the [Program - Settings](#) section. By default, C:\PageGateData\Logs\
- 5) Next, we'll need to set the end of record delimiter but let's go in to what that means first. When the UDP Server is receiving information, it doesn't really know what in the data is important; it's just receiving UDP packets of data. The end of record delimiter tells the UDP server that the expressed series of characters denotes a break in the data stream.

PageGate Admin: How to Configure

PageGate

For example, the UDP server could receive a series of 5 packets:

- a) This is
- b) a
- c) test message #
- d) This is a new
- e) message

If you set `EndOfRecordDelimiter = #`, that tells the UDP server that everything before every `#` received is a data block to be parsed.

Put another way, the end of record delimiter tells the UDP server how to break down the information it's receiving in to processable chunks.

It's the "over" of a CB radio conversation.

5) Below that, there are `RequiredText` and `RequiredTextOperator` values that can be set.

`RequiredText` specifies text that must appear for a data block to be processed and you may specify multiple values by separating each with a `|` (pipe).

The `RequiredTextOperator` specifies whether you're using OR or AND logic.

For example, let's say you specify the following:

```
RequiredText=alarm|notification
```

```
RequiredTextOperator=AND
```

That tells the UDP server that both the words `alarm` AND `notification` need to appear in a received data block for that data block to be treated as a message to send.

As another example:

```
RequiredText=alarm|alert
```

```
RequiredTextOperator=OR
```

That tells the UDP server that either the words `alarm` or `alert` need to appear in a received data block for that data block to be treated as a message to send.

6) Next we have `EndOfRecordTimeout`, whose value is expressed in milliseconds. This value tells the UDP server how long to wait before timeout after receiving data, to accept a data record.

By default, this is set to 3000 milliseconds or 3 seconds.

7) `RejectOnTimeout` determines whether timeouts should post a rejection notice or not. By default, this is set to true.

PageGate Admin: How to Configure PageGate

6

8) AcceptResponse determines what the UDP server will echo to the sending system upon acceptance of a data block.

9) RejectResponse determines what the UDP server will echo to the sending system upon rejection of a data block.

10) Last, we have two optional settings: StaticRecipient and StaticSender

StaticRecipient allows you to specify the name of a specific recipient or group in PageGate that ALL traffic from the UDP server should be passed through.

StaticSender allows you to hard code a sender's name that should be associated with ALL traffic from the UDP server.

11) Save the .ini file and close notepad.

Step 3: Run the UDP Server

1) Open a command prompt with elevated permission.

2) Change directory to the UDP Server's program directory. By default, c:\PageGateData\UDP\

3) Type the following and hit enter:
pgudpserver /install

4) Exit the command prompt.

5) Open the Services list in Windows.

6) Go in to the Properties of the PageGate UDP Server service.

7) Change the Startup Type to Automatic and Start the Service.

Step 4: Configure the GetASCII API to process the UDP Server's messages

1) Go to Interfaces - GetASCII - Settings.

2) Set the 'Polling Directory' to the output folder. By default, C:\PageGateData\UDP\

3) Check Enabled.

4) Click Apply.

PageGate Admin: How to Configure

PageGate

5) Go to Program - Settings.

6) In the 'Run on this server' section, check GetASCII

7) Click Apply.

8) If prompted for credentials, leave them blank and click Apply again.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the UDP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc and it is a Standard file type.

TAP to ASCII

This interface is a companion module for GetTAP and unlocks the ability to monitor additional serial/RS232 ports or modems, up to as many as you need. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#). This API requires the GetASCII API to function.

Step 1: Create a directory for the TAP to ASCII receiver

1) In Windows, create a new directory for the TAP to ASCII receiver. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called TAP2ASCII X where X is the number of the interface running as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\TAP2ASCII1\).

2) Unzip the TAP to ASCII file you were sent upon purchasing the API in to the folder you created in step 1.

Step 2: Configure the TAP to ASCII receiver's settings

1) The first thing we'll need to do is modify the TAP2ASCII.ini file. Open the file in notepad.

To give you a little more information, the TAP2ASCII.ini file controls the behavior of the TAP to ASCII receiver.

2) Change RunStyle=Application to RunStyle=Service

3) Change SwitchStyle=False to SwitchStyle=True

4) Set DataFolder= to the directory you created in step 1. By default, DataFolder=C:\PageGateData\TAP2ASCII1\

PageGate Admin: How to Configure PageGate

6

5) Set ComPort= to the serial or modem port that should be monitored.

6) Set ComSettings= to the communication parameters required. Values must be passed as baud rate,parity,databits,stopbits

For example, this string would use 9600 baud, no parity, 8 data bits and 1 stop bit:
ComSettings=9600,n,8,1

and this string would use 1200 baud, even parity, 7 data bits and 1 stop bit:
ComSettings=1200,e,7,1

7) Set InitString= to the word 'direct' for RS232/serial connections. Example,
InitString=direct

Set InitString= to the appropriate AT command string to initialize your modem in 'answer' mode. For most modems, you can use ATS0=1 to configure the modem to answer after 1 ring. Example, InitString=ATS0=1

8) Save the .ini file and close notepad.

Step 3: Run the TAP to ASCII receiver

1) Open a command prompt with elevated permission.

2) Change directory to the TAP to ASCII receiver's program directory. By default, c:\PageGateData\TAP2ASCII1\

3) Type the following and hit enter:
TAP2ASCII1 /install

4) Exit the command prompt.

5) Open the Services list in Windows.

6) Find and start the TAP 2 ASCII 1 service.

Step 4: Configure the GetASCII API to process the TAP to ASCII receiver's messages

Note: You must configure a basic polling directory before specifying these advanced polling settings.

1) Go to Interfaces - GetASCII - Settings - Advanced Polling - Files.

2) Click Add.

PageGate Admin: How to Configure PageGate

6

3) Specify the following:

Recipient: *

From: Leave blank or specify a value

File(s): *.t2a

Path: Specify the output path of the TAP to ASCII receiver. By default, c:\PageGateData\TAP2ASCII1\

File Type: Standard

4) Click Apply, then click Apply again.

TAP IP Server

The TAP IP Server allows PageGate to receive the TAP protocol by TCP request instead of serial or modem line. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#). This API requires the GetASCII API to function.

Step 1: Create a directory for the TAP IP Server

1) In Windows, create a new directory for the TAP IP receiver. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called IPTAP as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\IPTAP\).

2) In Windows, browse in to the PageGate [program directory](#).

3) Move the TapIpServer.exe, pininfo.txt and tapip.ini files to the folder you created in step 1.

Step 2: Configure the TAP IP Server's settings

1) The first thing we'll need to do is modify the tapip.ini file. Open the file in notepad.

To give you a little more information, the tapip.ini file controls the behavior of the TAP IP Server. If you are running the evaluation version, it will stop itself every 3-5 days, and/or after 500 messages, whichever comes first.

2) Change the OutputFolder= value to the directory you created in step 1. By default, C:\PageGateData\IPTAP\

PageGate Admin: How to Configure PageGate

6

3) Change the LogFolder= value to your [logging directory](#). If you are unsure of what this folder is, open the PageGate Admin and go to the [Program - Settings](#) section. By default, C:\PageGateData\Logs\

4) Save the .ini file and close notepad.

5) Open pininfo.txt in notepad.

This file tells the SNPP server what range of numbers can be passed to the server. The format of the entries defined in this text file are:

```
"Starting PIN", "Ending PIN", "recipient type"
```

If you would like ID numbers in a range to be recognized as the recipient's name (not using [ad-hoc](#), but using numeric recipient names in PageGate), use the text "PG_Recipient" as shown below:

```
"0001","9999", "PG_Recipient"
```

If you would like ID numbers in a range to be passed through PageGate as intact ID's to another source in an [ad-hoc](#) range, use the name of a [Recipient in PageGate whose type is Ad-Hoc](#).

```
"0000001","4999999","adhoc1"
```

To have it accept all potential alphanumeric values, use the following statements:

```
"*", "*", "PG_Recipient"
```

```
"*", "*", "adhocrecipient"
```

6) After declaring the appropriate ID ranges, save and close the file.

Step 3: Register the TAP IP Server

Without its registration, the TAP IP Server can be run in a demo mode where it will stop the service/shut itself off every 3 to 5 days or every 100 messages, whichever happens first.

To register the TAP IP Server:

1) Open the tapip.ini file in notepad.

2) Add the following line to the bottom of the file:

```
Registration= Your PageGate version 10 registration key number
```

3) Save and close the tapini.ini file.

Step 4: Run the TAP IP Server

1) Open a command prompt with elevated permission.

PageGate Admin: How to Configure PageGate

6

- 2) Change directory to the TAP IP Server's program directory. By default, c:\PageGateData\IPTAP\

```
cd c:\PageGateData\IPTAP
```
- 3) Type the following and hit enter:

```
tapipserver /install
```
- 4) Exit the command prompt.
- 5) Open the Services list in Windows.
- 6) Go in to the Properties of the PageGate TAP IP Server service.
- 7) Change the Startup Type to Automatic and Start the Service.

Step 5: Configure the GetASCII API to process the TAP IP Server's messages

- 1) Go to Interfaces - GetASCII - Settings.
- 2) Set the 'Polling Directory' to the output folder. By default, C:\PageGateData\IPTAP\

```
cd c:\PageGateData\IPTAP
```
- 3) Check Enabled.
- 4) Click Apply.
- 5) Go to Program - Settings.
- 6) In the 'Run on this server' section, check GetASCII
- 7) Click Apply.
- 8) If prompted for credentials, leave them blank and click Apply again.

Note: If you are already using the GetASCII module, you can use the [Advanced Polling Files](#) section to specify the TAP IP Server's output folder as a secondary polling location. The File(s) value will need to be set to *.asc and it is a Standard file type.

Serial to ASCII

This interface is a companion module for GetSerial and unlocks the ability to monitor additional serial/RS232 ports (physical or virtual), up to as many as you need. Messages received by this API are output to a directory as a text file that is then read in by the [GetASCII API](#). This API requires the GetASCII API to function.

Step 1: Create a directory for the Serial to ASCII receiver

PageGate Admin: How to Configure PageGate

6

1) In Windows, create a new directory for the Serial to ASCII receiver. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called S2ASCII_X where X is the number of the interface running as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\S2ASCII1\).

2) Unzip the Serial to ASCII file you were sent upon purchasing the API in to the folder you created in step 1.

Step 2: Configure the Serial to ASCII receiver's settings

1) The first thing we'll need to do is modify the Serial2ASCII.ini file. Open the file in notepad.

To give you a little more information, the Serial2ASCII.ini file controls the behavior of the Serial to ASCII receiver.

2) Change RunStyle=Application to RunStyle=Service

3) Change SwitchStyle=False to SwitchStyle=True

4) Set DataFolder= to the directory you created in step 1. By default, DataFolder=C:\PageGateData\S2ASCII1\

5) Set ComPort= to the serial/RS232 port that should be monitored.

6) Set ComSettings= to the communication parameters required. Values must be passed as baud rate,parity,databits,stopbits

For example, this string would use 9600 baud, no parity, 8 data bits and 1 stop bit:

ComSettings=9600,n,8,1

and this string would use 1200 baud, even parity, 7 data bits and 1 stop bit:

ComSettings=1200,e,7,1

7) The behavior of the [Symbols](#), [Record](#), [Recipient](#), [Sender](#) and [Message](#) sections follows that of the GetSerial module. [Click here](#) for more information.

8) After configuring all appropriate input parameters, save and close the .ini file.

Step 3: Run the Serial to ASCII receiver

PageGate

- 1) Open a command prompt with elevated permission.
- 2) Change directory to the Serial to ASCII receiver's program directory. By default, c:\PageGateData\S2ASCII1\
`\PageGateData\S2ASCII1\`
- 3) Type the following and hit enter:
`Serial2ASCII1 /install`
- 4) Exit the command prompt.
- 5) Open the Services list in Windows.
- 6) Find and start the Serial 2 ASCII 1 service.

Step 4: Configure the GetASCII API to process the TAP to ASCII receiver's messages

Note: You must configure a basic polling directory before specifying these advanced polling settings.

- 1) Go to Interfaces - GetASCII - Settings - Advanced Polling - Files.
- 2) Click Add.
- 3) Specify the following:
Recipient: *
From: Leave blank or specify a value
File(s): *.s2a
Path: Specify the output path of the TAP to ASCII receiver. By default, c:\PageGateData\Serial2ASCII1\
File Type: Standard

- 4) Click Apply, then click Apply again.

Filter Pack

The filter pack is a scripting system that allows you to modify any input in to or output from any of PageGate's interfaces based on nearly any criteria of your choosing.

Things you can do with the filtering system:

- Conditionally modify who a message is intended to go to based on key words or key phrases.
 - All fire, rescue and accident calls should be dispatched but non-emergency calls should not be.
- Only alarms with "CRITICAL" in them should be delivered.

PageGate Admin: How to Configure PageGate

- Messages with the keyphrase "HAZARD" should go to a high priority group instead of the normal distribution list.
 - Replies with the word 'subscribe' should perform one action but replies with the word 'unsubscribe' should perform another.
 - Modify the contents or structure of data to turn it in to something more user-friendly and readable.
 - Pull content from an API (like the National Weather Service, a Ticket Tracking system like Jira or ServiceNow, etc)
- Exam
ples:
- Parse XML, HTML, JSON or any other form of delimited text in to readable message content
 - Trigger a Powershell or other script.
 - This allows you to have PageGate trigger other applications or processes by tying them to a conditional filter in PageGate. Powershell is incredibly powerful and with the right script, you could have PageGate execute functions that would modify another application's database, perform network or domain lookup functions or anything else made available by Powershell.
- Descr
iption:

This is certainly not a comprehensive list of things you could do and, effectively, this filter allows you to modify any input fed in to or output delivered from PageGate based on nearly any selection criteria of your choosing.

Over the course of this documentation, we'll be going over how to configure, code and implement PageGate's filtering system.

First, the filter scripts require the file 'pgfilter.exe' to reside in the PageGate program directory and will need to reference an XML file containing the script, which can reside in any directory the PageGate server has access to (whether that's on the local hard drive or on a network share via UNC path). You should receive all necessary files for the filtering system upon purchase but if you do not have them, please contact our sales and registration department at 781-829-0500 x 1 or by email at sales@notepage.com.

Second, it's important to understand the chain of events that [triggers the script to run](#).

For more detailed information, please see: [How to Implement the Script](#)

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

PageGate

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ASCII value. The ASCII value of ~ is: \126</p>

Note: PageGate's [filter script](#) supports [RegEx expressions](#), which can allow you to implement an incredibly intricate system.

How to Implement the Script

Step 1: Copy the filter's executable in to the PageGate program directory

The file 'pgfilter.exe' must reside in the PageGate program folder. If you do not have a copy of this file, please contact our sales and registration department at sales@notepage.com or by phone at 781-829-0500.

Step 2: Determine what you want filtered

Filter scripts are called through PageGate's template system and can be broken down in to two categories: Inbound and Outbound

Inbound filters can trigger on and modify content passed to the program. These filters are applied to the program's input interfaces and on a global level to effect all input interfaces.

Outbound filters can trigger on and modify messages delivered by the program. These filters can be applied to a recipient, carrier, connector and on a global level.

You can have a system reference multiple filters for different purposes at the same time.

For example, you could configure an inbound filter to strip XML and JSON formatting from received data as well as a global outbound filter that inserted a custom header before every message. You could then make exceptions to that global rule by configuring unique filters for a carrier, recipient or connector.

PageGate Admin: How to Configure PageGate

6

You could also configure an SMS reply processing system that filtered reply messages containing certain keywords to specific recipients and groups while also implementing an outbound filter to customize the contents or formatting of messages delivered to them.

You could do everything described all at once. It really is just a matter of determining what you want to do, then implementing an appropriate solution.

Step 3: Write the filter

Each filter exists as an XML file in a directory of your choosing; this directory just needs to be accessible to the PageGate server, whether that's on the local hard drive or via UNC to a network path. The XML file containing your filter may also be called anything you like, you can give the file any name as long as its extension is .xml

[Steps for writing filters can be found here.](#)

Step 4: Reference the filter

After writing a filter, you'll need to add reference it in the 'Settings' section of a template in PG Admin.

Inbound/Interface Filters

Value	Description	Example
InterfaceFilter	<p>This applies a filter to processed data. This allows the interfaces to apply the initial layer of data sorting and collection and then applies the filter to the result.</p> <p><u>GetAscii</u> Standard polling: Filter is applied against temp file with recipient, sender and message</p> <p>Advanced polling: Filter is applied against temp file with recipient, sender and message, and after lookup substitution</p> <p><u>GetMail</u> Filter is applied against temp file with recipient, sender and message, after they have been extracted from the email, and decoding has taken place</p> <p><u>GetSerial</u> Filter is applied against temp file with recipient, sender and message, and after lookup substitution</p>	InterfaceFilter = c:\PageGateData\Filters\yourfilter.xml

	<p><u>GetTap</u> Filter is applied against each message that is received. The file processed by PGFilter will have Id as first line, second line blank, and message on remaining lines.</p> <p><u>GetWeb</u> Filter is applied against temp file with recipient, sender and message</p> <p>This can be used on an interface or global level. If used in an interface, it applies the filter to that interface. If used on a global level, it applies the filter to all input from all interfaces.</p>	
RawInterfaceFilter	<p>This applies a filter to raw, unprocessed data. This applies the filter before the interface has a chance to read the data. The specific behavior varies depending on the interface in question.</p> <p><u>GetAscii</u> Standard polling: Filter is applied against the raw text file before GetAscii tries to process it</p> <p>Advanced polling: Filter is applied against each record that is found, before it is processed any further (record pattern matching done first)</p> <p><u>GetMail</u> Filter is applied against the raw file received from the PageGate mail server process. All headers are intact and no decoding has taken place yet. Envelope values for recipient and sender are not available for modification in the raw mail file (because they are read from the envelope).</p> <p><u>GetSerial</u> Filter is applied against each record found before it is processed any further (record pattern matching is done first)</p>	RawInterfaceFilter = c:\PageGateData\Filters\yourfilter.xml

	<p><u>GetTap</u> Filter is applied against each message that is received.</p> <p><u>GetWeb</u> Filter is applied against raw text file created by webgate.exe CGI program</p> <p>This can be used on an interface or global level. If used in an interface, it applies the filter to that interface. If used on a global level, it applies the filter to all input from all interfaces.</p>	
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Outbound Filters

Value	Description	Example
Filter	<p>This applies a filter to outbound messages. This can be applied on a recipient, carrier, connector or global level.</p> <p>If used in a recipient template, all messages delivered to that recipient will be processed through the filter.</p> <p>If used in a carrier template, all messages delivered through that carrier will be processed through the filter.</p> <p>If used in the program template, it applies to all output from the program.</p>	<p>ConnectorFilter = c: \PageGateData\Filters\yourfilter.xml</p>
Connector Filter	<p>This applies a filter to the messages delivered by a connector. This filter can only modify the contents of a message and cannot modify the recipient or sender information.</p> <p>This can be applied on a connector or global level.</p> <p>If used in the program template, it applies to all connectors in the program.</p>	<p>Filter = c: \PageGateData\Filters\yourfilter.xml</p>

SubjectFilter	<p>This applies a filter to the subject line of any outbound email. This value is only referenced by the SMTP protocol and is not referenced by other protocols.</p> <p>This can be applied on a recipient, carrier, connector or global level.</p> <p>If used in a recipient template, all SMTP/email messages delivered to that recipient will have their subject lines processed through the filter.</p> <p>If used in a recipient template, all SMTP/email messages delivered to through that SMTP/email carrier will have their subject lines processed through the filter.</p> <p>If used in the program template, all SMTP/email traffic sent from the system will have its subject line processed through the filter.</p>	SubjectFilter = c:\PageGateData\Filters\yourfilter.xml
---------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------

Hierarchy

An Interface (and Global) template can have both a RawInterfaceFilter and an InterfaceFilter. Both filters will run in that order if they are both set.

Connectors will run any outbound filter according to which template has the highest priority. In order, they are:

- 1) Recipient
- 2) Carrier
- 3) Connector
- 4) Global

After implementing a filter script, it is highly recommend that you go through a [debugging process](#) to verify what you've configured. See [Debugging the Script](#) for more information.

Debugging the Script

To debug the filter script, you'll first need to set the [Debug tag](#) to true.

By setting Debug to True, that puts the script in to debugging mode. When in debugging mode, the script will display a window that shows the result of every component of your filter. It will display whether the filter performed its functions successfully or if there was an error and, if there was an error, what the error was related to. For example, "Match not found" in the event that your search criteria couldn't be found.

PageGate Admin: How to Configure PageGate

6

This makes tracking down errors in the script much easier than it otherwise would be.

If you're using `preproc.bat`, rename it to `preprocv.bat`

If you're using `pstproc.bat`, rename it to `pstprocv.bat`

The `v` in the file name stands for Verbose, so you're effectively putting the script in to a mode that will display far more information on the screen.

The Filter Script

Filters can be simple or complex depending on what you want to do. The most basic filters are search and replace functions, where you remove specific patterns like XML tags from text or replace short-codes with more verbose information. The more advanced filters can conditionally modify who receives a message, what the message says and execute functions via batch file or Powershell script, among other things.

The filters themselves are written in XML and, to start, you may want to look at [the complete filter script template](#). That will give you a general overview of each section and all of the possible values you could use in them.

To break things down, there is one major section, Filter, that determines what each section of the script will do. Additionally, there are two conditional sections, InputFile and OutputFile, plus an optional debugging mode:

- [Filter](#)

The Filter section determines what you want to do with the data read as input. A single filter script may reference multiple individual `<Filter>` sections to perform multiple tasks within the same data set. It's also important to note that the filter system supports the use of Regex, allowing for a high level of customization and control.

- [InputFile](#)

Most scripts will not need this parameter. When using the filter system in conjunction with the GetAscii interface or a FILE protocol carrier, the InputFile section determines which file(s) you want the filter to scan.

- [OutputFile](#)

Most scripts will not need this parameter. When using the filter system in conjunction with the GetAscii interface or a FILE protocol carrier, the OutputFile section determines what you want to name the files output by the filter's modifications.

- [Debug](#)

The Debug section determines whether you want a debug level review of the filter to appear before the filter completes its actions. This is incredibly useful when first

PageGate

configuring a script and we recommend that this flag be enabled until the filter is ready to put in to production.

For most scripts, you'll want to start with the [Filter](#) section.

The Complete Filter Script Template

Note: The InputFile and OutputFile sections are not required for most scripts. For more information, please see [The Filter Script](#).

```
<InputFile>
  <Name>
    (Parameter), static input filename
  </Name>
  <Delete>
    True, (False) - deletes the Input File, even if <Abort> is used
  </Delete>
</InputFile>
<OutputFile> - not used if the .dll version of the filter is running the script
  <Name>
    (InputFile), static output filename
  </Name>
</OutputFile>
<Debug>
  <Console>
    <Active>
      True, (False)
    </Active>
  </Console>
  <File>
    <Active>
      True, (False)
    </Active>
    <Name>
      debug log filename
    </Name>
    <Append>
      True, (False) - appends to existing file instead of recreating it
    </Append>
  </File>
  <LogInputFile>
    True, (False)
  </LogInputFile>
  <LogOutputFile>
    True, (False)
```

PageGate Admin: How to Configure PageGate

```
</LogOutputFile>
```

```
</Debug>
```

```
<Filter>*
```

```
  <Comment>
```

optional comment text that will be shown in debug output

```
  </Comment>
```

```
  <FilterName>
```

If you are using the <Goto> tag for looping, then the filter you want to jump back to must have a name

```
  </FilterName>
```

```
  <Lookup> - if the Lookup section is included, no other sections are processed (i.e.
```

```
<MatchFound> and <MatchNotFound> are not run
```

```
  <Variable>
```

(All), Recipient, Sender, Message, Temp1-10

```
  </Variable>
```

```
  <CaseSensitive>
```

True, (False)

```
  </CaseSensitive>
```

```
  <SpaceSensitive>
```

True, (False) - If False, leading and trailing spaces are ignored

(otherwise they need to match exactly)

```
  </SpaceSensitive>
```

```
  <List>
```

A list of comma separated values - one per line
or a filename that contains the list

If the text in the <Variable> text matches the first value of a pair in the

list,

the second value in the pair is substituted for it

Multiple values can be included in the first value by separating them

with | symbols

Values can be quoted - if so use two quotes in a row (""") to include a single quote in the value's text

If no match is found, and the last pair in the list have a * for it's first value, then this will be used as a default value

Comments: any line of text with a pound sign (#) as the very first

character

(no leading spaces or tabs) will be considered a comment,

and ignored.

```
  </List>
```

```
</Lookup>
```

```
<Search>
```

```
  <Type>
```

(Literal), RegEx, GreaterThan, LessThan - for GreaterThan and

LessThan, <Variable> is GreatherThan/LessThan <Text>

```
  </Type>
```


PageGate Admin: How to Configure PageGate

6

```
<Variable>
    (All), Recipient, Sender, Message, Temp1-10, Flag
</Variable>
<Text>*
    search text, True, False
    must be quoted to retain beginning and/or ending whitespace
</Text>
</Search>
<MatchFound> - if you don't include a <Search> section,
<Replace> - multiple Replace sections are allowed
    <Type>
        (Literal), RegEx, And, Or - And/Or sets Flag equal to Flag
And/Or <Text>
    </Type>
    <Variable>
        (All), Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Scope>
        (Match), Entire - replace only the matched text, or the entire
variable's contents
    </Scope>
    <ReplaceAll>
        True, (False) - replaces all matches in the variable
    </ReplaceAll>
    <Text>*
        replacement text, True, False
        must be quoted to retain beginning and/or ending whitespace
    </Text>
</Replace>
<WriteFile>
    <Name>*
        if match found, filename to save to - most macros are valid
    </Name>
    <Variable>
        (All), Recipient, Sender, Message, Temp1-10
    </Variable>
</WriteFile>
<Run>
    <Command>*
        run this command (no spaces in folders or filenames)
    </Command>
    <Async>
        (True), False - if true, the filter continues immediately, if false it
waits for
```

PageGate Admin: How to Configure PageGate

6

the program it started to finish

```
</Async>
<Show> - true give the program a window to run in, false hides the
program
    True, (False)
</Show>
</Run>
<RunFilter>
  <SettingsFile>*
    path and name of new filter settings file
  </SettingsFile>
  <Variable>
    ( ), All, Recipient, Sender, Message, Temp1-10
    name of the variable that holds the text to be sent to the filter
    * if Variable is used (and not empty), the InputFile value is
ignored
    </Variable>
  <InputFile>
    path and name of input file - ignored if Variable section is set
  </InputFile>
  <DebugFile>
    path and name of debug file
  </DebugFile>
  <AppendDebug>
    True, (False)
  </AppendDebug>
  <UseDll>
    True, False - if pgfilter.dll is running, then default value is True,
otherwise it is False
  </UseDll>
</RunFilter>
<ReadFile>
  <Name>*
    if match found, filename to read - most macros are valid
  </Name>
  <Variable>
    (All), Recipient, Sender, Message, Temp1-10
  </Variable>
  <Delete>
    True, (False) - delete the file after reading it
  </Delete>
</ReadFile>
<Goto>
  Name of filter to jump to - *** be careful not to make infinite loops
```

PageGate Admin: How to Configure PageGate

6

```
</Goto>
<Abort>
    True, (False) - exit without writing to the Output File (which is the Input
File by default)
</Abort>
<Continue>
    (True), False - whether or not to continue processing the next filter
block
</Continue>
<MatchFound>
<MatchNotFound> - this section is always run if there is no <Search> section
<Replace> - multiple Replace sections are allowed
    <Type>
        (Literal), And, Or - And/Or sets Flag to Flag And/Or
replacement Text
    </Type>
    <Variable>
        (All), Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Text>*
        replacement text, True, False
        must be quoted to retain beginning and/or ending whitespace
    </Text>
</Replace>
<WriteFile>
    <Name>*
        if match found, filename to save to - most macros are valid
    </Name>
    <Variable>
        (All), Recipient, Sender, Message, Temp1-10
    </Variable>
</WriteFile>
<Run>
    <Command>*
        run this command (no spaces in folders or filenames)
    </Command>
    <Async>
        (True), False - if true, the filter continues immediately, if false it
waits for
        the program it started to finish
    </Async>
    <Show> - true give the program a window to run in, false hides the
program
    True, (False)
```

PageGate Admin: How to Configure PageGate

```

</Show>
</Run>
<RunFilter>
  <SettingsFile>*
    path and name of new filter settings file
  </SettingsFile>
  <Variable>
    (, All, Recipient, Sender, Message, Temp1-10
    name of the variable that holds the text to be sent to the filter
    * if Variable is used (and not empty), the InputFile value is
ignored
    </Variable>
  <InputFile>
    path and name of input file - ignored if Variable section is set
  </InputFile>
  <DebugFile>
    path and name of debug file
  </DebugFile>
  <AppendDebug>
    True, (False)
  </AppendDebug>
  <UseDll>
    True, False - if pgfilter.dll is running, then default value is True,
otherwise it is False
  </UseDll>
</RunFilter>
<ReadFile>
  <Name>*
    if match found, filename to read - most macros are valid
  </Name>
  <Variable>
    (All), Recipient, Sender, Message, Temp1-10
  </Variable>
  <Delete>
    True, (False) - delete the file after reading it
  </Delete>
</ReadFile>
<Goto>
  Name of filter to jump back to - be careful not to make infinite loops
</Goto>
<Abort>
  True, (False) - exit without writing to the Output File (which is the Input
File by default)
</Abort>

```

PageGate Admin: How to Configure PageGate

6

```
<Continue>
    (True), False - whether or not to continue processing the next filter
block
    </Continue>
    <MatchNotFound>
</Filter>
<Filter>...
```

Notes:

Sections with * are required

Values in () are default values if section is not present

The first commandline parameter is the name of the InputFile to be processed
If an <InputFile><Name> section is used, its value overrides the commandline
parameter value

The default settings file name is pgfilter.xml

The settings file name can be set by using settingsfile= as a parameter on
the commandline. The input file name would then be the next parameter

The debug file name can be set by using debugfile= as a parameter on
the commandline. The input file name would then be the next parameter

By default, when writing debug info to a file, the file is recreated each time
Using the commandline parameter debugappend=true will append to the file
instead

*** WARNING *** don't leave debugappend set for long periods, because the file
will keep growing forever

OutputFile, ReadFile, WriteFile support variables (in addition to the normal ones):

```
%InputFilePathAndName%
%InputFilePath%
%InputFileName%
%InputFileNameBase%
%InputFileNameExtension%
%FilterIndex%
```

Section values for Recipient, Sender, Message, and All really just mean:

Recipient: first line of text,

Sender: second line of text,

Message: third and remaining lines of text

All: the entire contents of the file

PageGate Admin: How to Configure PageGate

Literal search text special characters:

- ~ as first character means the start of the variable's contents
 - ~ as last character means the end of variable's contents
 - ? matches a single character
 - * matches anything (including nothing)
 - \ following 3 digits are a decimal character value
- the ~, ?, *, and \ characters must use \ with decimal equivalent to be used

literally

RegEx replacement:

- Backreferences are allowed in replacement text (i.e. \1 \2 \3 ...)
- \r, \n, and \### (3 digits) literals are supported for inserting CR, LF, and

ASCII

Literal search text variables

- %Recipient% - the current value for Recipient
- %Sender% - the current value for Sender
- %Message% - the current value for Message
- %All% - the current value for All of the text
- %Temp1-10% - the current value for that Temp variable
- \ following 3 digits are a decimal character value

Literal replacement text variables:

- %Recipient% - the current value for Recipient
- %Sender% - the current value for Sender
- %Message% - the current value for Message
- %All% - the current value for All of the text
- %Match% - the actual text that matched the search text (Literal or RegEx match types)
- %Group1-10% - RegEx group backreferences obtained from a RegEx search
- %Temp1-10% - the current value for that Temp variable
- %InputFileName% - the name of the input file being processed
- %Year%, %Month%, %Day%, %MHour%, %Hour%, %Minute%, %Second%, %MSecond%, %AMPM%
- %Unique% - unique number - guaranteed unique to the computer - pretty unique otherwise
- \ following 3 digits are a decimal character value

Functions:

- Format is %FunctionName(input-value)% - it will be replaced with the function's output value
- %DateTimeToUnixTime()%

PageGate Admin: How to Configure

PageGate

The default value of the Temp1-10 variables are an empty string

The default value of the Flag variable is false

Comments: The <Comment> section can be used in each <Filter> and it will be output

in the debug log, or any line of text with a pound sign (#) as the very first character (no leading spaces or tabs) will be considered a comment, and ignored, and also not logged

The Recipient, Sender, Message and ALL variables are dynamically linked together. So, for

example, if you change the Sender, the first line of All will also be updated.

Changing

All will update the Recipient, Sender and Message parts

When the order of things matter:

<Filter> block are executed in the order they appear (unless <Goto> is used)

In a <Filter> block:

<Search> (if it exists) runs first and determines if the <MatchFound> or <MatchNotFound> block is run

<Replace> <WriteFile> <Run> <RunFilter> <ReadFile> <GoTo>

<Abort> <Continue> run in that order

There can be multiple <Replace> <WriteFile> <Run> <RunFilter> <ReadFile> blocks and they will run

in the order they appear - i.e. all <Replace> blocks will run, then all <WriteFile> blocks, etc.

RegEx uses a Pearl compatible implementation syntax

good reference: <http://www.regular-expressions.info/tutorialcnt.html>

Breaking changes

<Section></Section> is renamed to <Variable></Variable>

In the <Replace> section the <All></All> tag has been renamed to

<ReplaceAll></ReplaceAll>

In the <Replace> <Scope> section the value: 'Section' is renamed to 'Entire'

All exiting <Replace> section have to be surrounded with <MatchFound></MatchFound> tags

Recipient, Sender, Message and All are now dynamically linked together

Bare Filter Template - for easier cutting/pasting

* in sections with multiple value options, the first in the list is the default value

PageGate Admin: How to Configure

PageGate

```

<InputFile>
  <Name>
    Parameter
  </Name>
  <Delete>
    False, True
  </Delete>
</InputFile>
<OutputFile>
  <Name>
    InputFile
  </Name>
</OutputFile>
<Debug>
  <Console>
    <Active>
      False, True
    </Active>
  </Console>
  <File>
    <Active>
      False, True
    </Active>
    <Name>

    </Name>
    <Append>
      False, True
    </Append>
  </File>
  <LogInputFile>
    False, True
  </LogInputFile>
  <LogOutputFile>
    False, True
  </LogOutputFile>
</Debug>
<Filter>
  <Comment>

  </Comment>
  <FilterName>

  </FilterName>

```


PageGate Admin: How to Configure PageGate

6

```
<Lookup>
  <Variable>
    All, Recipient, Sender, Message, Temp1-10
  </Variable>
  <CaseSensitive>
    False, True
  </CaseSensitive>
  <SpaceSensitive>
    False, True
  </SpaceSensitive>
  <List>

  </List>
</Lookup>
<Search>
  <Type>
    Literal, RegEx, GreaterThan, LessThan
  </Type>
  <Variable>
    All, Recipient, Sender, Message, Temp1-10, Flag
  </Variable>
  <Text>

  </Text>
</Search>
<MatchFound>
  <Replace>
    <Type>
      Literal, RegEx, And, Or
    </Type>
    <Variable>
      All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Scope>
      Match, Entire
    </Scope>
    <ReplaceAll>
      False, True
    </ReplaceAll>
    <Text>

    </Text>
  </Replace>
  <WriteFile>
```

PageGate Admin: How to Configure PageGate

```

<Name>

</Name>
<Variable>
    All, Recipient, Sender, Message, Temp1-10, Flag
</Variable>
</WriteFile>
<Run>
    <Command>

    </Command>
    <Async>
        True, False
    </Async>
    <Show>
        False, True
    </Show>
</Run>
<RunFilter>
    <SettingsFile>

    </SettingsFile>
    <Variable>
        , All, Recipient, Sender, Message, Temp1-10
    </Variable>
    <InputFile>

    </InputFile>
    <DebugFile>

    </DebugFile>
    <AppendDebug>
        False, True
    </AppendDebug>
    <UseDll>
        True, False
    </UseDll>
</RunFilter>
<ReadFile>
    <Name>

    </Name>
    <Variable>
        All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>

```

PageGate Admin: How to Configure PageGate

6

```
<Delete>
    False, True
</Delete>
</ReadFile>
<Goto>

</Goto>
<Abort>
    False, True
</Abort>
<Continue>
    True, False
</Continue>
<MatchFound>
<MatchNotFound>
<Replace>
    <Type>
        Literal, And, Or
    </Type>
    <Variable>
        All, Recipient, Sender, Message, Temp1-10, Flag
    </Variable>
    <Text>

    </Text>
</Replace>
<WriteFile>
    <Name>

    </Name>
    <Variable>
        All
    </Variable>
</WriteFile>
<Run>
    <Command>
    </Command>
    <Async>
        True, False
    </Async>
    <Show>
        False, True
    </Show>
</Run>
```

PageGate Admin: How to Configure PageGate

6

```
<RunFilter>
  <SettingsFile>

  </SettingsFile>
  <Variable>
    , All, Recipient, Sender, Message, Temp1-10
  </Variable>
  <InputFile>

  </InputFile>
  <DebugFile>

  </DebugFile>
  <AppendDebug>
    False, True
  </AppendDebug>
  <UseDll>
    True, False
  </UseDll>
</RunFilter>
<ReadFile>
  <Name>

  </Name>
  <Variable>
    All, Recipient, Sender, Message, Temp1-10, Flag
  </Variable>
  <Delete>
    False, True
  </Delete>
</ReadFile>
<Goto>

  </Goto>
  <Abort>
    False, True
  </Abort>
  <Continue>
    True, False
  </Continue>
<MatchNotFound>
</Filter>
```

PageGate Admin: How to Configure

Filter PageGate

The Filter section determines what you want to do with the data read as input. A single filter script may reference multiple individual <Filter> sections to perform multiple tasks within the same data set. You can also find [filter statement examples here](#) and [complete script examples here](#).

```
<Filter>
  <Search>
    See Search section
  </Search>
  <Replace>
    See Replace section.
  </Replace>
  <Continue>
    See Values and Description
  </Continue>
  <Abort>
    See Values and Description
  </Abort>
</Filter>
```

Flag	Values and Description
Search	Please review the Search section for a detailed explanation of this tag and its components.
Replace	Please review the Replace section for a detailed explanation of this tag and its components.
Continue	This tag determines whether or not you want the script to continue to the next action if it finds a match. This tag can be set to a value of True or False. For example, let's say we're searching for the text 345 within the input file 's data. Let's also say that no match for 345 can be found in the input data. If Continue is set to True, the filter will move to the next specified <Filter> section. If Continue is set to False, the script will immediately terminate upon not finding a match.
Abort	If the search section finds a match and Abort has a value of True, the script is aborted. Basically, the input data is left unchanged and no more filter blocks are processed. It just abruptly ends the filter processing and quits out without saving anything.

PageGate

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>

The Search section determines what you want the filter to look for.

```

<Search>
  <Section>
    See Values and Description
  </Section>
  <Type>
    See Values and Description
  </Type>
  <Text>
    See Values and Description
  </Text>
</Search>
    
```

Flag	Values and Description	
Section	This tag determines where the filter should look for the data specified in the Text tag (see below).	
	Value s	Explanation
	All	All tells the filter to search the entire data set read from the input file.
	Recipient	Recipient tells the filter to search only in the first line of the file.

PageGate

	<table border="1"> <tr> <td data-bbox="363 197 483 275">Sender</td> <td data-bbox="483 197 1463 275">Sender tells the filter to search only in the second line of the file.</td> </tr> <tr> <td data-bbox="363 275 483 352">Message</td> <td data-bbox="483 275 1463 352">Message tells the filter to search only in the third and following lines of the file.</td> </tr> <tr> <td data-bbox="363 352 483 558">Temp</td> <td data-bbox="483 352 1463 558">Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags. See Example 1 and Example 3.</td> </tr> <tr> <td data-bbox="363 558 483 678">Flag</td> <td data-bbox="483 558 1463 678">Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains. See Example 1 and Example 3.</td> </tr> </table>	Sender	Sender tells the filter to search only in the second line of the file.	Message	Message tells the filter to search only in the third and following lines of the file.	Temp	Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags. See Example 1 and Example 3 .	Flag	Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains. See Example 1 and Example 3 .
Sender	Sender tells the filter to search only in the second line of the file.								
Message	Message tells the filter to search only in the third and following lines of the file.								
Temp	Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags. See Example 1 and Example 3 .								
Flag	Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains. See Example 1 and Example 3 .								
Type	<p>Values: Literal or RegEx</p> <p>This tag determines whether you want to use literal text or a RegEx statement in the Text tag.</p>								
Text (Literal)	<p>If the Type tag is set to Literal, this tells the filter to search on the text entered in to the Text tag. For example, you could enter the literal text Fire Department if you wanted the filter to key off of the literal text "Fire Department" (without the quotes).</p> <p>This tag supports the following boolean parameters:</p> <ul style="list-style-type: none"> * - Multiple Character Wildcard For example, fire* would find fire, firewood, fireplace, fireman and similar. ? - Single Character Wildcard For example, fire? would find fires, fired but would not find fire. ~ as first character means start of section/file ~ as last character means end of section/file <p>For example, ~12345~ would only read 12345. Without the ~ before and after the string, the filter would recognize 12345 in 0123456.</p> <p>This tag also supports the following variables:</p> <ul style="list-style-type: none"> %Recipient% - the current value for Recipient %Sender% - the current value for Sender %Message% - the current value for Message %All% - the current value for All of the text %Temp% - the current value for the Temp variable \ - following 3 digits are a decimal character value. See Below. 								
Text (Regex)	<p>The PageGate Filter script supports RegEx expressions. RegEx uses a Perl compatible implementation syntax and a good reference guide for the language can be found here:</p>								

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;">IMPORTANT</p> ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output. This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.

The Replace section determines what you want to do with the data found in the [Search](#) section.

```
<Replace>
  <Section>
    See Values and Description
  </Section>
  <Type>
    See Values and Description
  </Type>
  <Scope>
    See Values and Description
  </Scope>
  <All>
    See Values and Description
  </All>
  <Text>
    replacement text goes here
    must be quoted to retain beginning and/or ending whitespace
    See Values and Description
  </Text>
</Replace>
```


Flag	Values and Description														
Section	<p>Values: All, Recipient, Sender, Message, Temp, Flag</p> <p>This tag determines which section the filter perform the replacement within.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Explanation</th> </tr> </thead> <tbody> <tr> <td>All</td> <td>All tells the filter to replace any instance of the searched text within the entire data set read from the input file.</td> </tr> <tr> <td>Recipient</td> <td>Recipient tells the filter to replace values only in the first line of the file.</td> </tr> <tr> <td>Sender</td> <td>Sender tells the filter to replace values only in the second line of the file.</td> </tr> <tr> <td>Message</td> <td>Message tells the filter to replace values only in the third and following lines of the file.</td> </tr> <tr> <td>Temp</td> <td>Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags.</td> </tr> <tr> <td>Flag</td> <td>Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains.</td> </tr> </tbody> </table>	Value	Explanation	All	All tells the filter to replace any instance of the searched text within the entire data set read from the input file.	Recipient	Recipient tells the filter to replace values only in the first line of the file.	Sender	Sender tells the filter to replace values only in the second line of the file.	Message	Message tells the filter to replace values only in the third and following lines of the file.	Temp	Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags.	Flag	Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains.
	Value	Explanation													
	All	All tells the filter to replace any instance of the searched text within the entire data set read from the input file.													
	Recipient	Recipient tells the filter to replace values only in the first line of the file.													
	Sender	Sender tells the filter to replace values only in the second line of the file.													
	Message	Message tells the filter to replace values only in the third and following lines of the file.													
	Temp	Temp is a string variable that can be passed strings of characters to hold, then use its value to search on later. Temp does not support boolean operators. For example, you can have certain information stored in the Temp value, then both search on and output that Temp value in successive Filter tags.													
	Flag	Flag is a boolean value that can only hold a true or false statement. However, you can run "And" or "Or" statements against it for advanced logic chains.													
Type	<p>Values: Literal, RegEx, And, Or</p> <p>This tag determines whether you want to use literal text or a RegEx statement in the Text tag. This tag can also be used as a situational boolean operator to determine how to proceed.</p> <p>The boolean values (Or & And) modify the Flag value. So, for example, you could have the filter search for a value and set it as the Flag, then have the replace filter use "Or" to indicate that the Flag value may be more than one thing. You could also use the And operator to tell the filter that the Flag value must contain multiple variables to be matched. See Example 1 and Example 3.</p>														
	Scope	Values: Match or Section													
All	Values: True or False														
Text (Literal)	<p>If the Type tag is set to Literal, this tells the filter to search on the text entered in to the Text tag. For example, you could enter the literal text Fire Department if you wanted the filter to key off of the literal text "Fire Department" (without the quotes).</p> <p>This tag supports the following boolean parameters:</p>														

PageGate

	<p>* - Multiple Character Wildcard For example, fire* would find fire, firewood, fireplace, fireman and similar.</p> <p>? - Single Character Wildcard For example, fire? would find fires, fired but would not find fire.</p> <p>~ as first character means start of section/file ~ as last character means end of section/file For example, ~12345~ would only read 12345. Without the ~ before and after the string, the filter would recognize 12345 in 0123456.</p>
Text (Reg ex)	The PageGate Filter script supports RegEx expressions. RegEx uses a Pearl compatible implementation syntax and a good reference guide for the language can be found here: http://www.regular-expressions.info/tutorialcnt.html

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;">IMPORTANT</p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>

To better demonstrate how some of the flags function, here are a few examples of complete filter statements and an explanation of what they do.

Here is an example of a [Literal Text Replacement Command](#).

Here is an example of an [Abort Command](#).

PageGate Admin: How to Configure

PageGate

This Filter statement tells the script to terminate if the [Flag](#) variable is set to False:

```
<Filter>
  <Search>
    <Section>
      Flag
    </Section>
  </Search>
  <Type>
    Literal
  </Type>
  <Text>
    False
  </Text>
</Filter>
<Replace>
  <Section>
    Flag
  </Section>
  <Scope>
    Section
  </Scope>
  <Type>
    And
  </Type>
  <Text>
    False
  </Text>
  <All>
    False
  </All>
</Replace>
<Continue>
  False
</Continue>
</Filter>
```

This Filter statement tells the script to replace one piece of literal text with another. More specifically, this statement tells the script to search for the literal text Boiler Room and replace it with boiler_tech_group.

```
<Filter>
  <Search>
    <Section>
      All
    </Section>
```

```

        <Type>
            Literal
        </Type>
        <Text>
            Boiler Room
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
            boiler_tech_group
        </Text>
        <All>
            False
        </All>
    </Replace>
</Filter>
InputFile

```

The InputFile section determines which file(s) you want the filter to scan.

```

<InputFile>
    <Name>
        See Values and Description
    </Name>
    <Delete>
        See Values and Description
    </Delete>
</InputFile>

```

Flag	Values and Description
Name	<p>Values: Parameter or Static Input File Name</p> <p>This tag determines the name(s) of the file(s) that you want the filter to modify. Just to note, the value Parameter is literal and not figurative. To use this value, you'll want to use the text Parameter, including the capital P. When the filter is</p>

PageGate

	<p>engaged by the pstproc.bat file, it is fed the file name being used by the filter. The Parameter value tells the filter that it should simply accept the file name as it was passed by the batch file.</p> <p>Alternatively, you can specify a static file for input. For example, c:\PageGateData\Filter\winsck9000.log would tell the filter to only read from a specific file named winsck9000.log. As another example, you could specify c:\PageGateData\Filter*.log and that would tell the filter to read all log files placed in the directory.</p>
Delete	<p>Values: True or False</p> <p>This tag determines whether you want the filter to delete the input file when it has finished processing it. If you do, set this tag to True. If you do not, set this tag to False.</p>

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p> <p>This character is used when you want to represent a character as represented by its ASCII Decimal value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.</p>

OutputFile

The OutputFile section determines what you want to name the files output by the filter's modifications. This section is defined as follows:

```
<OutputFile>
  <Name>
```

PageGate Admin: How to Configure PageGate

See Values and Description

</Name>

</OutputFile>

Flag	Values and Description
Name	<p>Values: InputFile or Static Input File Name</p> <p>This tag determines the names of the files that you want the filter to output after it has modified the data read from the input file.</p> <p>The value InputFile (this is case sensitive) tells the filter that the file output should mirror the original input file's name. So, for example, if the file read was named example.txt, then the file output would also be called example.txt</p> <p>Alternatively, you can specify a static naming convention for the output files. This value also supports the following variables:</p> <ul style="list-style-type: none"> %InputFileName% %InputFileNameBase% %InputFileNameExtension% %InputFilePathAndName% %InputFilePath% <p>So, for example, you could specify this as the value: c:\PageGateData\ASCII\%InputFileNameBase%.asc</p> <p>That would tell the filter that it should use the original name of the file read but the output file should have the extension .asc instead of the extension on the original file.</p>

As an important note, due to the way this script is programmed, certain characters are treated as command characters instead of literal text characters. The following characters represent certain command functions in the script:

Character	Function
~	~ as first character means start of section/file, ~ as last character means end of section/file. For example, ~123~ would only find 123 and not 01234 or 1237, etc. ~123 would find 12340 but would not find 01234. 123~ would find 0123 but would not find 01234.
?	Single character wildcard.
*	Multiple character wildcard.
\	<p style="text-align: center;"><u>IMPORTANT</u></p> <p>ASCII value prefix (Ex: \013 for a carriage return, \010 for a line feed). The \ must be followed by a three digit code to indicate the character to be output.</p>

PageGate

This character is used when you want to represent a character as represented by its [ASCII Decimal](#) value. For example, since the filter directly references the character ~ in its function, if you want to use the ~ character in your text, you'll need to represent it with its ascii value. The ascii value of ~ is \126.

Debug

The Debug section determines whether you want a debug level review of the filter to appear before the filter completes its actions. This is incredibly useful when first configuring a script and we recommend that this flag be enabled until the filter is ready to put in to production. This section is defined as follows:

```
<Debug>
    See Values and Description
</Debug>
```

Flag	Values and Description
Debug	<p>Values: True or False</p> <p>This tag determines whether you want to review the debug level information for the filter before it completes its actions. Set this flag to True to review the debugging information, set this flag to False if you do not.</p>

Please see [Debugging the Script](#) for more information.

Complete Script Examples

Due to the complex nature of the Filter script, we've created a few examples of how scripts can be implemented. This is by no means a complete list of everything that the script can do and is intended to be a set of examples to guide you through the possibilities the script offers.

[Example 1](#)

In this example, we've configured the filter script to search the message read from the input file for the text BIOHAZARD and replace it with bio_hazard. We're then inserting the static text WARNING! EXTREME HAZARD DETECTED: before the matched search string.

[Example 2](#)

In this example, we've configured the filter to scan through XML tags for the text Technical or Network in the <AlertType> tag, then pull in information from the <Incident> and <Description> tags. This collated data is then being passed to a statically named group in PageGate: techsupervisors.

[Example 3](#)

PageGate

In this example, we've configured the filter to scan for certain keywords and key phrases. If it finds them, messages containing that text should not be allowed to go through and should instead be filtered.

[Example 4](#)

In this example, we've configured the filter script for a dispatching agency that needs all fire, rescue and accident calls dispatched but does not want non-emergency calls dispatched. To do this, the script reads in the entire input file and looks for certain flags like FIRE,DISPATCHED or RESCUE,DISPATCHED or ACCID,DISPATCHED.

If it finds any of those three strings, the script then begins massaging and reformatting the data to remove unwanted information and to determine who the message should go to and what the message should say.

In this particular example, all fire and rescue calls need to be delineated from each other so that the fire calls go to the fire department and the rescue calls go to the rescue department but all accident calls should go to both fire and rescue departments.

[Example 5](#)

In this example, we're using an [HTTPGET](#) carrier to poll data from the National Weather service to have it delivered as an SMS.

[Example 6](#)

This example is designed to work with an [SMS Replies](#) system and is configured to look for the words 'subscribe' and 'unsubscribe' in the body of reply messages. If it finds 'subscribe', it reconfigures the message to go to the recipient or group in PageGate named 'subscribe'. If it finds 'unsubscribe', it reconfigures the message to go to the recipient or group named 'unsubscribe'.

[Example 7](#)

This example is designed to work with an [SMS Replies](#) system and is configured so that any time someone replies to a message with 'unsubscribe' or 'subscribe', those replies execute a batch file or script contained within a batch file. This can be especially useful if you're using an SQL environment and have PageGate's database linked to it with ODBC expressions. In theory, you could code a script to be executed that would automatically remove an entry from the users table of the database any time the 'unsubscribe' script was executed.

[Example 8](#)

This example is designed to work with an [SMS Replies](#) system. This script is configured with a list of 'allowed' phone numbers and is also configured to scan the sender value of all replies. If the sender's phone number isn't in the 'allowed' list, the message is filtered as spam.

PageGate Admin: How to Configure

PageGate

Example 1

In this example, we've configured the filter script to search the message read from the input file for the text 'BIOHAZARD'. If it is found then the script will replace the current Recipient with 'bio_hazard' to indicate that all messages containing the word BIOHAZARD should go to the bio_hazard group. We're then inserting the static text 'WARNING! EXTREME HAZARD DETECTED:' before the matched search string.

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

This section of the filter configures the script to look for a static file named Example.txt in the same directory as the script and, when the script completes, it will not delete the original file and will output to a static file named Output.txt to the directory in which the script resides.

```
<InputFile>
  <Name>
    Example.txt
  </Name>
  <Delete>
    False
  </Delete>
</InputFile>
```

```
<OutputFile>
  <Name>
    Output.txt
  </Name>
</OutputFile>
```

```
<Debug>
  False
</Debug>
```

```
# This section of the filter configures the script to search
# for BIOHAZARD and if found, replaces the Recipient
# with bio_hazard.
```

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
```

PageGate Admin: How to Configure PageGate

```

    <Type>
      Literal
    </Type>
    <Text>
      BIOHAZARD
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Type>
      Literal
    </Type>
    <Scope>
      Section
    </Scope>
    <Text>
      bio_hazard
    </Text>
  </Replace>
  <Continue>
    True
  </Continue>
</Filter>

```

This section of the filter configures the script to insert
 # WARNING! EXTREME HAZARD DETECTED: before the BIOHAZARD text
 <Filter>

```

  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      BIOHAZARD
    </Text>
  </Search>
  <Replace>
    <Section>
      Message
    </Section>

```

PageGate Admin: How to Configure PageGate

```

<Type>
    Literal
</Type>
<Scope>
    Match
</Scope>
<Text>
    WARNING! EXTREME HAZARD DETECTED: %Match%
</Text>
</Replace>
</Filter>

```

This is an example of the input file for [filter example 1](#). This is what it looks like before the filter engages.

```

supervisors
AlarmMonitor
BIOHAZARD ROOM 354

```

Another example:

```

it_alerts
AlarmMonitor
BIOHAZARD RULES IN EFFECT FOR WING 2

```

When [the data from example 1](#) is put through [the example 1 filter](#), this is the output:

```

bio_hazard
AlarmMonitor
WARNING! EXTREME HAZARD DETECTED: BIOHAZARD ROOM 354

```

or like this:

```

bio_hazard
AlarmMonitor
WARNING! EXTREME HAZARD DETECTED: BIOHAZARD RULES IN EFFECT FOR
WING 2

```

Example 2

Example 2

In this example, we've configured the filter to scan through an XML file for the text 'Technical' or 'Network' in the <AlertType> tag. It will then temporarily store the information from the <Incident> and <Description> tags in the Temp variable. This collected data is then being passed to a statically named group in PageGate: techsupervisors.

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

```

<InputFile>

```

PageGate Admin: How to Configure PageGate

```

<Name>
    Parameter
</Name>
<Delete>
    True
</Delete>
</InputFile>

<Debug>
    False
</Debug>

<OutputFile>
    <Name>
        c:\PageGateData\ASCII\%InputFileNameBase%.asc
    </Name>
</OutputFile>

```

This section of the Filters tells the script to look for
any lines that have <AlertType>Technical</AlertType> and
set the Flag variable to True if it is found

```

<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            <AlertType>Technical</AlertType>
        </Text>
    </Search>
    <Replace>
        <Section>
            Flag
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Or
        </Type>
        <Text>

```

PageGate Admin: How to Configure PageGate

```

    True
  </Text>
</Replace>
</Filter>

```

This section of the Filters tells the script to look for any
lines that have the <AlertType>Network</AlertType> and
set the Flag variable to True if it is found

```

<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      <AlertType>Network</AlertType>
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Or
    </Type>
    <Text>
      True
    </Text>
  </Replace>
</Filter>

```

This Filter section tells the script to abort if no instances of
the two search strings can be found:

```

<Filter>
  <Search>
    <Section>
      Flag
    </Section>
    <Type>

```

PageGate Admin: How to Configure PageGate

```

    Literal
    </Type>
    <Text>
        False
    </Text>
</Search>
<Replace>
    <Section>
        Flag
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        And
    </Type>
    <Text>
        False
    </Text>
</Replace>
<Continue>
    False
</Continue>
</Filter>

```

This section of the filter tells the script to grab data
contained in the <Incident> tag.

```

<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            <Incident>*</Incident>
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>

```

PageGate Admin: How to Configure PageGate

```

    <Scope>
      Section
    </Scope>
  <Type>
    Literal
  </Type>
  <Text>
    %Match%
  </Text>
</Replace>
</Filter>

```

This section of the filter tells the script to grab data from
the <Description> tag

```

<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      <Description>*</Description>
    </Text>
  </Search>
  <Replace>
    <Section>
      Temp
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      %Temp% %Match%
    </Text>
  </Replace>
</Filter>

```

This section of the filter removes the tag

PageGate Admin: How to Configure PageGate

<Incident> from the message.

```
<Filter>
  <Search>
    <Section>
      Temp
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      <Incident>
    </Text>
  </Search>
  <Replace>
    <Section>
      Temp
    </Section>
    <Scope>
      Match
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      </Text>
    </Text>
  </Replace>
</Filter>
```

This section of the filter removes the tag
</Incident> from the message.

```
<Filter>
  <Search>
    <Section>
      Temp
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      </Incident>
    </Text>
  </Search>
```


PageGate Admin: How to Configure PageGate

```

    <Replace>
      <Section>
        Temp
      </Section>
    </Scope>
  </Scope>
</Type>
  Literal
</Type>
<Text>

</Text>
</Replace>
</Filter>

# This section of the filter removes the tag
# <Description> from the message.

```

```

<Filter>
  <Search>
    <Section>
      Temp
    </Section>
  <Type>
    Literal
  </Type>
  <Text>
    <Description>
  </Text>
</Search>
  <Replace>
    <Section>
      Temp
    </Section>
  <Scope>
    Match
  </Scope>
  <Type>
    Literal
  </Type>
  <Text>

  </Text>

```

PageGate Admin: How to Configure PageGate

```
</Filter>
```

This section of the filter removes the tag
</Description> from the message.

```
<Filter>
  <Search>
    <Section>
      Temp
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      <Incident>
    </Text>
  </Search>
  <Replace>
    <Section>
      Temp
    </Section>
    <Scope>
      Match
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      </Text>
    </Text>
  </Replace>
</Filter>
```

This Filter tells the script to output the file in the standard
PageGate GetASCII file format for direct processing:

```
<Filter>
  <Search>
    <Section>
      Flag
    </Section>
    <Type>
      Literal
```

PageGate Admin: How to Configure PageGate

```

</Type>
  <Text>
    True
  </Text>
</Search>
<Replace>
  <Section>
    All
  </Section>
  <Scope>
    Section
  </Scope>
  <Type>
    Literal
  </Type>
  <Text>
    TechSupervisors\013\010CAD\013\010%Temp%
  </Text>
</Replace>
</Filter>

```

This is example XML content to be processed by [filter example 2](#). This is what it looks like before the filter engages.

```

<XML>
  <AlertType>Technical</AlertType>
  <Requester>Station 3</Requester>
  <Assets>PC01, PC02</Assets>
  <Category>Desktop Support</Category>
  <Incident>Replace Computers</Incident>
  <Description>Workstations 01 and 02 should be replaced on Thursday</Description>
</XML>

```

Another example:

```

<XML>
  <AlertType>Network</AlertType>
  <Requester>Admin Group</Requester>
  <Assets>Router</Assets>
  <Category>Network Support</Category>
  <Incident>DNS Failure</Incident>
  <Description>DNS failures detected in node 2</Description>
</XML>

```

When [the data from filter example 2](#) is processed by [filter example 2](#), this is the output

PageGate Admin: How to Configure

PageGate

content

techsupervisors

CAD

Replace Computers, Workstations 01 and 02 should be replaced on Thursday

Another example:

techsupervisors

CAD

DNS Failure, DNS failures detected in node 2

Example 3

In this example, we're scanning for keywords and keyphrases that indicate a message shouldn't be sent. If the filter finds any of the keywords or keyphrases, it will filter those messages from going through. For this example, we've configured the filter to look for the key phrases "Return from Alarm" and "Alarm Acknowledged" as well as potential partial matches.

```
<InputFile>
  <Name>
    Parameter
  </Name>
  <Delete>
    True
  </Delete>
</InputFile>

<Debug>
  True
</Debug>

<OutputFile>
  <Name>
    c:\PageGateData\output\%InputFileNameBase%.asc
  </Name>
</OutputFile>

<Filter>
<Comment>look for Return From Alarm</Comment>
<Search>
  <Section>
    All
  </Section>
  <Type>
```

PageGate Admin: How to Configure PageGate

```

    </Type>
    <Text>
        Return From Alarm
    </Text>
</Search>
<Replace>
<Text>
</Text>
</Replace>
<Abort>
    True
</Abort>
</Filter>

<Filter>
<Comment>look for Partial Phrases of Return From Alarm</Comment>
<Search>
    <Section>
        All
    </Section>
    <Type>
        Literal
    </Type>
    <Text>
        Retu*
    </Text>
</Search>
<Replace>
<Text>
</Text>
</Replace>
<Abort>
    True
</Abort>
</Filter>

<Filter>
<Comment>look for Alarm Block Acknowledged</Comment>
<Search>
    <Section>
        All
    </Section>
    <Type>

```

PageGate Admin: How to Configure PageGate

```

    </Type>
    <Text>
        Alarm Block Acknowledged
    </Text>
</Search>
<Replace>
<Text>
</Text>
</Replace>
<Abort>
    True
</Abort>
</Filter>

<Filter>
<Comment>look for Partial "Alarm Block Acknowledged" phrases</Comment>
<Search>
    <Section>
        All
    </Section>
    <Type>
        Literal
    </Type>
    <Text>
        Block
    </Text>
</Search>
<Replace>
<Text>
</Text>
</Replace>
<Abort>
    True
</Abort>
</Filter>

```

Example 4

Example 3

In this example, we've configured the filter script for a dispatching agency that needs all fire, rescue and accident calls dispatched but does not want non-emergency calls dispatched. To do this, the script reads in the entire input file and looks for certain keywords like FIRE,DISPATCHED or RESCUE,DISPATCHED or ACCID,DISPATCHED.

PageGate Admin: How to Configure

6

PageGate

If it finds any of those three strings, the script then reformats the data, removes unwanted information, and determines who the message should be sent to.

In this particular example, all fire and rescue calls need to be separated from each other so that the fire calls go to the fire department and the rescue calls go to the rescue department, but all accident calls should go to both fire and rescue departments.

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

This section of the filter configures the script to accept whatever file name is passed by the preproc.bat file, to delete the input file when it's finished and to output the reformatted data to the c:\PageGateData\ASCII\ folder with a file name mirroring the original file but with a .asc extension instead of the original file's extension.

```
<InputFile>
  <Name>
    Parameter
  </Name>
  <Delete>
    True
  </Delete>
</InputFile>

<Debug>
  False
</Debug>

<OutputFile>
  <Name>
    c:\PageGateData\ASCII\%InputFileNameBase%.asc
  </Name>
</OutputFile>
```

```
# The next three sections of the Filters tells the script to look for any lines that have
# the text ,FIRE,DISPATCHED| or ,RESCUE,DISPATCHED| or ACCID,DISPATCHED| in
the input file
```

```
# and sets the temporary Flag variable to True if any are found
```

```
# look for lines with station FIRE,DISPATCHED identifier
```

```
<Filter>
  <Search>
    <Section>
      All
```

PageGate Admin: How to Configure PageGate

```

</Section>
  <Type>
    Literal
  </Type>
  <Text>
    ,FIRE,DISPATCHED|
  </Text>
</Search>
<Replace>
  <Section>
    Flag
  </Section>
  <Scope>
    Section
  </Scope>
  <Type>
    Or
  </Type>
  <Text>
    True
  </Text>
</Replace>
</Filter>

# look for lines with station RESCUE,DISPATCHED identifier
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,RESCUE,DISPATCHED|
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>

```


PageGate Admin: How to Configure PageGate

```

    <Type>
      Or
    </Type>
    <Text>
      True
    </Text>
  </Replace>
</Filter>

# look for lines with station ACCID,DISPATCHED identifier
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,ACCID,DISPATCHED|
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Or
    </Type>
    <Text>
      True
    </Text>
  </Replace>
</Filter>

```

This section test the value of the Flag variable
 # and aborts out of the script if it is not set to True
 # Note that the values in the <Replace> section really
 # don't matter because the script is aborting

abort if FIRE/RESCUE/ACCID DISPATCHED not found

PageGate Admin: How to Configure

PageGate

```

<Filter>
  <Search>
    <Section>
      Flag
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      False
    </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      And
    </Type>
    <Text>
      False
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

```

Now that the filter has determined that there is relevant data
 # in the message file, these filter statements will search for
 # department names and store what is found in the Temp variable

look for lines with station F-BAY identifier

```

<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>

```

PageGate Admin: How to Configure PageGate

6

```
<Text>
    /F-BAY
</Text>
</Search>
<Replace>
    <Section>
        Temp
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        Literal
    </Type>
    <Text>
        F-BAY
    </Text>
</Replace>
</Filter>

# look for lines with station F-STU identifier
<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            /F-STU
        </Text>
    </Search>
    <Replace>
        <Section>
            Temp
        </Section>
        <Scope>
            Section
        </Scope>
        <Type>
            Literal
        </Type>
        <Text>
```

PageGate Admin: How to Configure PageGate

```

F-STU
    </Text>
  </Replace>
</Filter>

```

```

# This Filter section tells the script to abort if no
# acceptable station is found within the data
# (the Temp variable is still blank)
<Filter>

```

```

  <Search>
    <Section>
      Temp
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      </Text>
  </Search>
  <Replace>
    <Section>
      Flag
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      And
    </Type>
    <Text>
      False
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

```

```

# This Filter section tells the script to grab a portion of the data read for modification:
# It grabs everything between the first character of the line and ,DISPATCHED| keyword
# and removes everything else
<Filter>
  <Search>

```

PageGate Admin: How to Configure PageGate

```

<Section>
    All
</Section>
</Type>
    Literal
</Type>
<Text>
    *,DISPATCHED|
</Text>
</Search>
<Replace>
    <Section>
        All
    </Section>
    <Scope>
        Section
    </Scope>
    <Type>
        Literal
    </Type>
    <Text>
        %Match%
    </Text>
</Replace>
</Filter>

```

This Filter flag takes the data read by the previous statement and modifies it:
 # It gets rid of extra stuff between date/time and address and replaces it with
 # a carriage return and line feed

```

<Filter>
    <Search>
        <Section>
            All
        </Section>
        <Type>
            Literal
        </Type>
        <Text>
            (Eastern Daylight Time)*!%Temp%,F,
        </Text>
    </Search>
    <Replace>
        <Section>
            All

```

PageGate Admin: How to Configure PageGate

```

</Section>
  <Scope>
    Match
  </Scope>
  <Type>
    Literal
  </Type>
  <Text>
    \013\010
  </Text>
</Replace>
</Filter>

```

The next two filter sections separate the rescue calls from the fire calls
 # so that they go to two different agencies/organizations/units.

```

<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      FIRE
    </Text>
  </Search>
  <Replace>
    <Section>
      Temp
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      %Temp%-FIRE
    </Text>
  </Replace>
</Filter>

<Filter>

```

PageGate Admin: How to Configure PageGate

```

<Search>
  <Section>
    All
  </Section>
</Search>
<Type>
  Literal
</Type>
<Text>
  RESCUE
</Text>
</Search>
<Replace>
  <Section>
    Temp
  </Section>
<Scope>
  Section
</Scope>
<Type>
  Literal
</Type>
<Text>
  %Temp%-RESCUE
</Text>
</Replace>
</Filter>

```

The next three Filter sections remove the text ,DISPATCHED| from the message
and replaces it with more readable text

get rid of DISPATCHED on RESCUE

```

<Filter>
  <Search>
    <Section>
      All
    </Section>
  </Search>
  <Type>
    Literal
  </Type>
  <Text>
    ,RESCUE,DISPATCHED|
  </Text>
</Search>
<Replace>

```

PageGate Admin: How to Configure PageGate

```

    <Section>
      All
    </Section>
  </Scope>
  Match
</Scope>
<Type>
  Literal
</Type>
<Text>
  \013\010RESCUE
</Text>
</Replace>
</Filter>

# get rid of DISPATCHED on FIRE
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,FIRE,DISPATCHED|
    </Text>
  </Search>
  <Replace>
    <Section>
      All
    </Section>
    <Scope>
      Match
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      \013\010FIRE
    </Text>
  </Replace>
</Filter>

```


PageGate Admin: How to Configure

PageGate

```
# get rid of DISPATCHED on ACCID
```

```
<Filter>
  <Search>
    <Section>
      All
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      ,ACCID,DISPATCHED|
    </Text>
  </Search>
  <Replace>
    <Section>
      All
    </Section>
    <Scope>
      Match
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      \013\010ACCID
    </Text>
  </Replace>
</Filter>
```

```
# This Filter section tells the script to output the file in the standard
# PageGate GetASCII file format for direct processing:
```

```
# convert to standard .asc format
```

```
<Filter>
  <Search>
    <Section>
      Flag
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      True
    </Text>
```

PageGate Admin: How to Configure PageGate

```

</Search>
  <Replace>
    <Section>
      All
    </Section>
    <Scope>
      Section
    </Scope>
    <Type>
      Literal
    </Type>
    <Text>
      %Temp%\013\010CAD\013\010Dispatch %Temp%\013\010%All%
    </Text>
  </Replace>
</Filter>

```

This is example content to be processed by [filter example 3](#). This is what it looks like before the filter engages.

```

2018-08-23 14:07:30.307 (Eastern Daylight Time) CISCOUNTSTAT||T=1|
V=6233M070737305319||T=4|V=02||T=0|V=U,00/F-BAY,P,PRIMARY STREET/CROSS
STREET,INJ ACCID,DISPATCHED||

```

When [the data from filter example 4](#) is processed by [filter example 4](#), this is the output content:

```

F-BAY
DISPATCH
P,PRIMARY STREET/CROSS STREET,INJ ACCID

```

Example 5

This filter script is designed to work with the National Weather service's XML content, which is generated by polling their web API.

```

<Debug>
  <Console>
    <Active>True</Active>
  </Console>
  <File>
    <Active>True</Active>
    <Name>weatherfilter.txt</Name>
    <Append>False</Append>
  </File>
  <LogInputFile>True</LogInputFile>
  <LogOutputFile>True</LogOutputFile>
</Debug>

```

PageGate Admin: How to Configure

PageGate

```

<Filter>
<Comment>Delete if no severe weather alerts</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>All</Variable>
    <Text><title>There are no active watches, warnings or
advisories</title></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Literal</Type>
      <Variable>All</Variable>
      <Scope>Entire</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Check to see if this is the raw weather data or the ascii formatted data. If the
latter, skip to Process.</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>Sender</Variable>
    <Text>WeatherSend</Text>
  </Search>
  <MatchFound>
    <Goto>Process</Goto>
  </MatchFound>
</Filter>

<Filter>
<Comment>Turn each entry block in to a separate file.</Comment>
<FilterName>EntryLoop</FilterName>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(?(s)(<entry>.*?</entry>)</Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>Temp8</Variable>
      <Scope>Entire</Scope>

```

PageGate Admin: How to Configure PageGate

6

```
<Text>YourRecipientOrGroup\013\010WeatherSend\013\010%
Group1%</Text>
  </Replace>
  <WriteFile>
    <Variable>Temp8</Variable>
    <Name>INSERT THE PATH TO YOUR WEATHER FOLDER\posts\
%InputFileNameBase%-%FilterIndex%.asc</Name>
  </WriteFile>
</MatchFound>
<MatchNotFound>
  <Replace>
    <Variable>All</Variable>
    <Scope>Entire</Scope>
    <Text></Text>
  </Replace>
  <Goto>Finished</Goto>
</MatchNotFound>
</Filter>

<Filter>
<Comment>Get rid of the entry block processed, then loop back until there are no more
entry blocks</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(s)<entry>.*?</entry></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
    <Goto>EntryLoop</Goto>
  </MatchFound>
  <MatchNotFound>
    <Goto>Finished</Goto>
  </MatchNotFound>
</Filter>

<Filter>
<FilterName>Process</FilterName>
<Comment>Grab cap:effective and put it in Temp4</Comment>
  <Search>
```

PageGate Admin: How to Configure PageGate

```

    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><cap:effective>(.*?)</cap:effective></Text>
</Search>
<MatchFound>
  <Replace>
    <Variable>Temp4</Variable>
    <Scope>Entire</Scope>
    <Text>%Group1%</Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Grab cap:expires and put it in Temp5, then put the final ID in
Temp6</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><cap:expires>(.*?)</cap:expires></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>Temp5</Variable>
      <Scope>Entire</Scope>
      <Text>%Group1%</Text>
    </Replace>
    <Replace>
      <Variable>Temp6</Variable>
      <Scope>Entire</Scope>
      <Text>%Temp4%%Temp5%</Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
  <Comment> Turn Temp6 in to a valid file name and put it in Temp7. It will be used
to read/write the requestId files</Comment>
  <Search>
    <Type>RegEx</Type>
    <Variable>Temp6</Variable>
    <Text>[^a-zA-Z0-9\-\.]</Text>
  </Search>
  <MatchFound>

```

PageGate Admin: How to Configure PageGate

```

    <Replace>
      <Type>RegEx</Type>
      <Variable>Temp7</Variable>
      <Scope>Entire</Scope>
      <Text>%Group1%</Text>
      <ReplaceAll>True</ReplaceAll>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Pull in the ID from the file</Comment>
  <MatchNotFound>
    <ReadFile>
      <Variable>Temp9</Variable>
      <Name>INSERT THE PATH TO YOUR WEATHER FOLDER\ids\%
Temp7%.asc</Name>
    </ReadFile>
  </MatchNotFound>
</Filter>

<Filter>
  <Comment>Check to see if the ID already exists. If it does, delete everything. If it
doesn't, write the ID file</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>Temp9</Variable>
    <Text>%Temp7%</Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>All</Variable>
      <Scope>Entire</Scope>
      <Text></Text>
    </Replace>
    <Goto>Finished</Goto>
  </MatchFound>
  <MatchNotFound>
    <WriteFile>
      <Variable>Temp7</Variable>
      <Name>INSERT THE PATH TO YOUR WEATHER FOLDER\ids\%
Temp7%.asc</Name>
    </WriteFile>
  </MatchNotFound>

```

PageGate Admin: How to Configure

PageGate

```

</Filter>
<Filter>
<Comment>Remove from cap:event to cap:certainty</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(s)<cap:event>.*?</cap:certainty></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <ReplaceAll>True</ReplaceAll>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

```

```

<Filter>
<Comment>Remove from entry to author</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(s)<entry>.*?</author></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <ReplaceAll>True</ReplaceAll>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

```

```

<Filter>
<Comment>Replace cap:areaDesc with LOCATION</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>All</Variable>
    <Text><cap:areaDesc></Text>

```

PageGate Admin: How to Configure PageGate

```

</Search>
  <MatchFound>
    <Replace>
      <ReplaceAll>True</ReplaceAll>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text>\013\010LOCATION:\032</Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Remove the XML tags from the link</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><link href="(.)"V></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text>%Group1%</Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Remove from cap:polygon to cap:entry</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text>(?)s<cap:polygon>.*?<Ventry></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <ReplaceAll>True</ReplaceAll>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

```


PageGate Admin: How to Configure PageGate

```

<Filter>
<Comment>Remove the summary</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><summary>.*</summary></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Remove the opening title tag</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text><title></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <Variable>All</Variable>
      <ReplaceAll>True</ReplaceAll>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>
<Filter>
<Comment>Remove the closing title tag</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text></title></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>

```

PageGate Admin: How to Configure PageGate

```

    <Variable>All</Variable>
    <ReplaceAll>True</ReplaceAll>
    <Scope>Match</Scope>
    <Text></Text>
  </Replace>
</MatchFound>
</Filter>

<Filter>
<Comment>Remove the closing cap:areaDesc tag</Comment>
  <Search>
    <Type>Regex</Type>
    <Variable>All</Variable>
    <Text></cap:areaDesc></Text>
  </Search>
  <MatchFound>
    <Replace>
      <Type>Regex</Type>
      <Variable>All</Variable>
      <ReplaceAll>True</ReplaceAll>
      <Scope>Match</Scope>
      <Text></Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>
<Comment>Clean up the text, part 1</Comment>
  <Search>
    <Type>Literal</Type>
    <Variable>All</Variable>
    <Text>\013\010\013\010</Text>
  </Search>
  <MatchFound>
    <Replace>
      <ReplaceAll>True</ReplaceAll>
      <Variable>All</Variable>
      <Scope>Match</Scope>
      <Text>\013\010</Text>
    </Replace>
  </MatchFound>
</Filter>

<Filter>

```

PageGate Admin: How to Configure

PageGate

```
<Comment>Processing completed</Comment>
<FilterName>Finished</FilterName>
  <MatchNotFound>
  </MatchNotFound>
</Filter>
```

Example 6

Example 5

This filter script is designed to work with an [SMS Replies](#) system and is configured to look for the words 'subscribe' and 'unsubscribe' in the body of reply messages. If it finds 'subscribe', it reconfigures the message to go to the recipient or group in PageGate named 'subscribe'. If it finds 'unsubscribe', it reconfigures the message to go to the recipient or group named 'unsubscribe'.

Note: The <InputFile> and <OutputFile> sections are not included, so the script will use the filename passed in by PageGate, and write the script's results back to the same filename when done.

Note: Lines prefaced with # are comment lines for description purposes. The filter ignore any lines that begin with #

look for 'unsubscribe' in message text (all lower case)

if found, route the message to 'unsubscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      unsubscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      unsubscribe-recipient-or-group-here
```

PageGate Admin: How to Configure PageGate

```

</Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

```

look for 'Unsubscribe' in message text (begins with capital)
 # if found, route the message to 'unsubscribe' recipient

```

<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Unsubscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      unsubscribe-recipient-or-group-here
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

```

look for 'subscribe' in message text (all lower case)
 # if found, route the message to 'subscribe' recipient

```

<Filter>
  <Search>
    <Section>
      Message
    </Section>

```

PageGate Admin: How to Configure PageGate

```

    <Type>
      Literal
    </Type>
    <Text>
      subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      subscribe-recipient-or-group-here
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

```

look for 'Subscribe' in message text (begins with capital)

if found, route the message to 'subscribe' recipient

```

<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>

```

PageGate Admin: How to Configure PageGate

```

<Text>
    subscribe-recipient-or-group-here
</Text>
</Replace>
<Continue>
    False
</Continue>
</Filter>

```

Example 7

This filter script is designed to work with an [SMS Replies](#) system and is configured so that any time someone replies to a message with 'unsubscribe' or 'subscribe', those replies will be sent to an 'unsubscribe' or 'subscribe' recipient in PageGate that will execute a batch file or script contained within a batch file.

To do so, there's a bit of preparation you'll need to do to tell PageGate what script to run. It's also important to note that this batch file can be used to execute powershell commands.

Step 1: Create a directory for your batch file

- 1) Create a directory for the batch file(s) to reside in. This will also be the output folder for the carrier we'll create in Step 2. For example, c:\PageGateData\unsubscribe\

Step 2: Create a FILE Carrier

- 1) Open PG Admin.
- 2) Right click Carriers, select Add.
- 3) Give the carrier a name, something straightforward like BATCH.
- 4) Set Protocol to: FILE
- 5) Set File Path to the directory you created in Step 1 above. By default, c:\PageGateData\unsubscribe\

Step 3: Create a recipient named 'unsubscribe' and tie it to the FILE carrier

- 1) Right click Recipients, select Add.
- 2) Name the recipient: unsubscribe
- 3) Set Carrier to the FILE carrier you created in step 2. By default, BATCH.
- 4) Click Apply.

Step 4: Place and edit the appropriate script files

PageGate Admin: How to Configure PageGate

- 1) Open the directory you created in step 1 above. By default, c:\PageGateData\unsubscribe\
- 2) Copy pstproc.bat from the PageGate program directory in to that folder.
- 3) Edit pstproc.bat to execute the desired effect.

Step 5: Configure the SMS Replies Filter System

look for 'unsubscribe' in message text
 # if found, route the message to 'unsubscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      unsubscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      unsubscribe-recipient
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>
```

look for 'Unsubscribe' in message text
 # if found, route the message to 'unsubscribe' recipient

```
<Filter>
  <Search>
    <Section>
      Message
```

PageGate Admin: How to Configure PageGate

```

</Section>
  <Type>
    Literal
  </Type>
  <Text>
    Unsubscribe
  </Text>
</Search>
<Replace>
  <Section>
    Recipient
  </Section>
  <Scope>
    Section
  </Scope>
  <Text>
    unsubscribe-recipient
  </Text>
</Replace>
<Continue>
  False
</Continue>
</Filter>

# look for 'subscribe' in message text
# if found, route the message to 'subscribe' recipient
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section

```


PageGate Admin: How to Configure PageGate

```

</Scope>
  <Text>
    subscribe-recipient
  </Text>
</Replace>
<Continue>
  False
</Continue>
</Filter>

# look for 'Subscribe' in message text
# if found, route the message to 'subscribe' recipient
<Filter>
  <Search>
    <Section>
      Message
    </Section>
    <Type>
      Literal
    </Type>
    <Text>
      Subscribe
    </Text>
  </Search>
  <Replace>
    <Section>
      Recipient
    </Section>
    <Scope>
      Section
    </Scope>
    <Text>
      subscribe-recipient
    </Text>
  </Replace>
  <Continue>
    False
  </Continue>
</Filter>

```

Example 8

This example is designed to work with an [SMS Replies](#) system. This script is configured with a list of 'allowed' phone numbers and is also configured to scan the sender value of all

PageGate

reply. If the sender's phone number isn't in the 'allowed' list, the message is filtered as spam.

```
<Filter>
```

```
<Comment>
```

This filter is scanning the sender value for a series of phone numbers. In this example, we're using a simple Regex string to create an 'or' statement. So, we're scanning for the first number or the second number or the third number and so on.

```
</Comment>
```

```
<Search>
```

```
<Section>
```

```
Sender
```

```
</Section>
```

```
<Type>
```

```
Regex
```

```
</Type>
```

```
<Text>
```

```
(phonenum1|phonenum2|phonenum3|phonenum4|etc)
```

```
</Text>
```

```
</Search>
```

```
<Replace>
```

```
<Comment>
```

If the filter finds any of the specified phone numbers, it assigns the intended delivery group or recipient.

```
</Comment>
```

```
<Section>
```

```
Recipient
```

```
</Section>
```

```
<Scope>
```

```
Section
```

```
</Scope>
```

```
<Text>
```

```
name of messaging group or recipient
```

```
</Text>
```

```
</Replace>
```

```
<Continue>
```

```
False
```

```
</Continue>
```

```
</Filter>
```

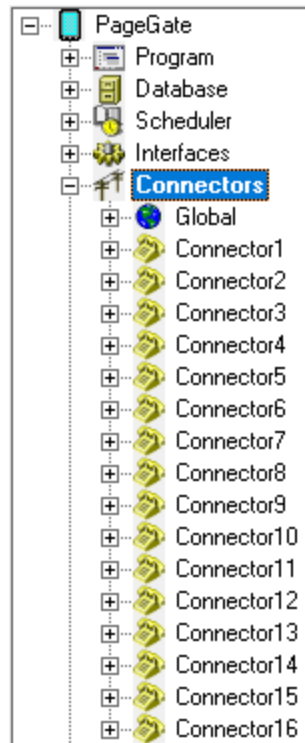
Connectors

PageGate can be broken down in to two general categories: Input and Output.

On the input side of things, we have [APIs](#) that allow PageGate to accept alerts, notifications, dispatches and messaging requests from a variety of sources.

PageGate

On the output side of things, we have the [Carriers](#) and Connectors. The [carriers](#) tell PageGate what communication method to use and the connectors are directly responsible for negotiating those communication sessions and the connectors actually perform the delivery.



Each connector in the program can handle any protocol at any time, so think of a connector as a pathway PageGate can use to deliver traffic. A connector can use a TCP based protocol like SMTP just as easily as it can use a cellular modem connected by RS-232 or USB.

- [Global](#)

This section configures the basic behavior of the connectors as well as displays the general status of all connectors.

- [Connector 1 through 16](#)

These sections control the communication settings of each connector.

In concept, a connector in PageGate is just a pathway to deliver messages as any connector in the program can, in most cases, handle any protocol at any time. So, you could implement a series of [carriers configured to use different delivery methods](#) and have all traffic processed through a single connector. However, each connector can only establish one message delivery sequence at a time. Also, each protocol has its own average negotiation time.

PageGate Admin: How to Configure PageGate

6

Protocol(s)	Average Negotiation Time Per Message
HTTPGET , HTTPPOST , AwsSNS , WCTP , File , RawTCP	>1 second
SNPP , Serial TAP , TAP over IP , XMPP , RawSerial	2 seconds
GSM-AT , GSM-AT-IP , SMTP	3 seconds
FaxSrv , Modem TAP , Fax , TONE , UCP	5+ seconds

So, as an example of how to work out the math on how many connectors you may need, let's say that you wanted to deliver 500 SMTP based messages.

At a delivery rate of 1 message every 3 seconds, it would take approximately 1500 seconds (or 25 minutes) for a single connector to deliver all 400 messages.

If you had two connectors, that would take approximately 750 seconds (12.5 minutes), if you had three, it would take approximately 500 seconds (8.33 minutes) and so on.

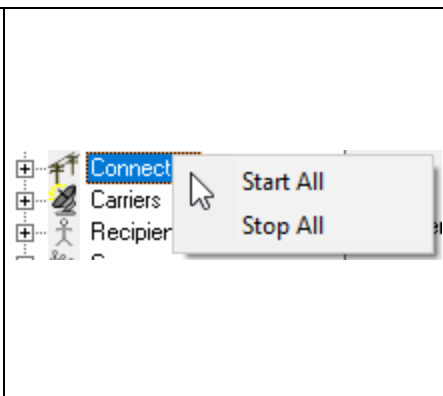
Right clicking on Connectors will provide a context menu with two options.

- [Start All](#)

Selecting this option will start all enabled Connectors, if stopped.

- [Stop All](#)

Selecting this option will stop the all enabled Connectors, if started.



Global

This section is where you can configure the global behavior of the connector system and see the general status of all connectors.



- [Settings](#)

This section controls the global behavior of the connector system.

- [Status](#)

This section displays the general status of all connectors.

Polling Interval	<input type="text" value="2"/>	(sec.)
Reply Check	<input type="text" value="60"/>	(sec.)
Msg Threshold	<input type="text" value="1"/>	
Retries	<input type="text" value="2"/>	
Retry Interval	<input type="text" value="10"/>	(sec.)
<input type="button" value="Apply"/>		

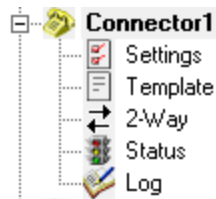
Setting	Function
Polling Interval	This field determines how often the connector system polls the database for updates to the messages table of the database from the Scheduler .
Reply Check	When implementing a cellular modem, router or gateway , this field determines how often the connector system should poll the modem for reply messages in memory.
Msg Threshold	<p>This field determines how many messages need to be in 'Pending' before the next connector in sequence is activated. Setting the Msg Threshold to 1 will evenly load balance all messages across all connectors.</p> <p>For example, let's say that you have the Dial Threshold set to 20 and that both Connectors 1 and 2 are enabled. Then let's say you queue 30 messages for PageGate to deliver. Connector 1 will first 20 and give the next 10 to Connector2 for processing.</p>
Retries	This determines the number of attempts the Connectors will make when delivering a message. If a message cannot be delivered after the specified number of retries, it is marked bad.
Retry Interval	This field determines how long each Connector waits between successive retries.
Apply	This button saves any changes recently made to the Global Settings of the Connectors.



If the status of a connector is stopped, it indicates that the connector is not running and this can be caused by a few things.

- The appropriate connector will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

Connectors



- [Settings](#)

This section controls the implementation of the selected connector

- [2-Way](#)

When implementing a [cellular modem, router or gateway](#), this section is used to poll the cellular device for replies.

- [Template](#)

PageGate Admin: How to Configure

6

PageGate

This section allows you to configure a custom template for messages processed by this connector.

- [Status](#)

This section displays the general status of the connector.

- [Log](#)

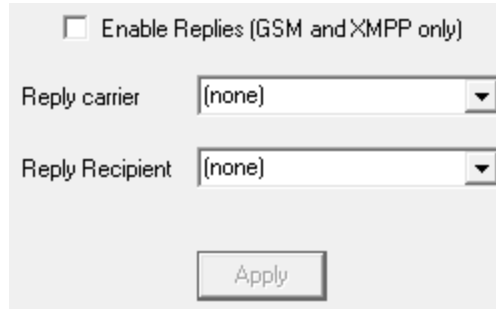
This section is a record of all activity performed by the connector.

Settings

The screenshot shows a settings form for a connector. The fields are: Connector (text box with 'Connector1'), Init String (dropdown menu with '(auto)'), Dialing Prefix (text box), Dedicated Carrier (dropdown menu with '(none)'), Serial Port (dropdown menu with '0'), and Direct Connect (checkbox). An 'Apply' button is located at the bottom of the form.

Setting	Function
Connector	This field tells you which Connector's settings you're currently viewing.
Init String	If this connector has been configured to use a modem, cellular modem, cellular router or cellular gateway, this value is passed to the device.
Dialing Prefix	If this connector has been configured to use a dial-up modem, this field determines any dialing prefix required to reach an external line.
Dedicated Carrier	This setting allows you to specify that this Connector can only handle messages from one carrier. This setting preempts the dialer threshold .
Serial Port	This setting is used if the Connector needs to be configured to use a modem, serial or USB connected cellular modem or to send data through a serial/RS232 connection. This field should reflect the COM port of the device you want to reference. A value of 0 tells PageGate that this connector is a should not be given any traffic for protocols that require serial/COM ports.
Direct Connect	This option should only be checked if the selected Connector is going to send information through a serial/RS232 connection.
Apply	This button saves all modifications made to the Connector settings.

Please review the [Two Way Messaging and Reply Processing](#) section of the documentation before enabling replies.



The screenshot shows a configuration panel with the following elements:

- A checkbox labeled "Enable Replies (GSM and XMPP only)" which is currently unchecked.
- A dropdown menu labeled "Reply carrier" with "(none)" selected.
- A dropdown menu labeled "Reply Recipient" with "(none)" selected.
- An "Apply" button at the bottom.

- **Enable Replies (GSM and XMPP only)**

Enabling this option will configure the connector to periodically check for messages from the cellular modem or XMPP server at an interval specified in the connector system's [Global](#) settings.

- **Reply carrier**

This carrier tells the reply system what language and protocol to use and should match the cellular hardware or XMPP server's carrier.

- **Reply Recipient**

This field determines who or what the reply messages should go to. For example, if all replies should go to a dispatch supervisor or a group of field techs, you can select them or their group as the reply recipient.

However, you can also implement a far more intricate system if you're licensed for the [GetASCII API](#) and there are many possible ways to implement this system.

First, it's important to note that the 2-way system can be fed through PageGate's [Filter Pack](#) to allow for conditional sending. For instance, with the filter pack, you can tell PageGate to conditionally change who receives the message based on what the text in the reply is. As an example, you could configure a filter system so that any time the word lunch shows up in a reply message, that reply is sent to the cafeteria group instead of the general replies group.

You could configure a system where replies from a certain phone number will always be copied to certain recipients or groups.

You could configure a system where certain keywords or phrases are replaced with other keywords or phrases. For example, you could translate a code like PD4N in to 4th Precinct.

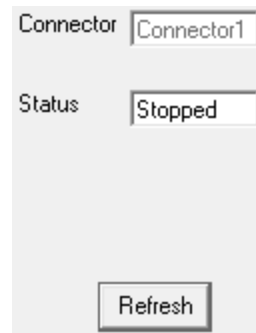
PageGate Admin: How to Configure PageGate

6

You could configure a system to describe every element I've just described all at once.

Please see the [Two Way Messaging and Reply Processing](#) section of the documentation for more information.

Status

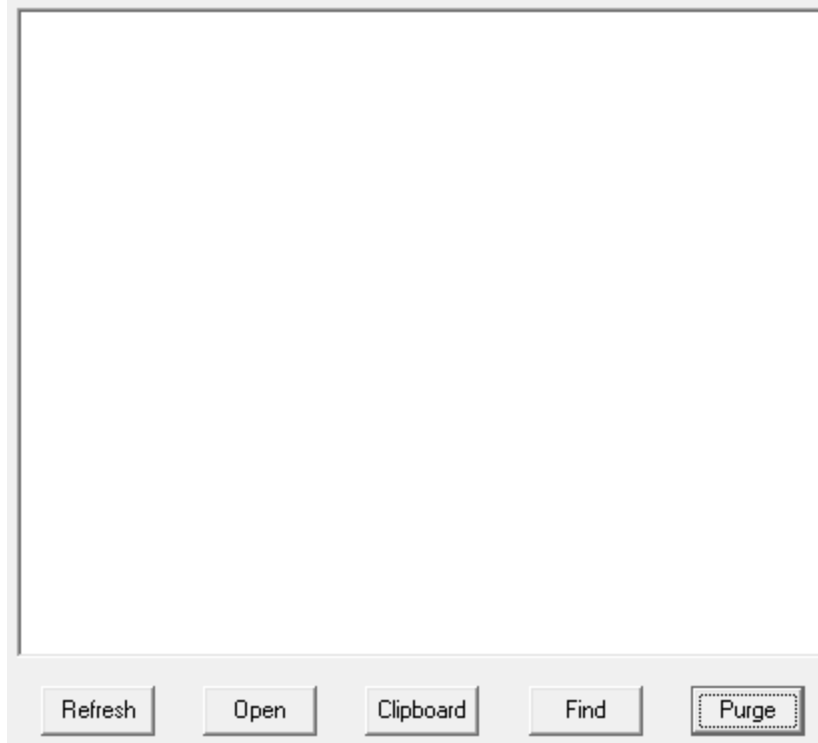


If the status of a connector is stopped, it indicates that the connector is not running and this can be caused by a few things.

- The appropriate connector will need to be checked in the ["Run on this server"](#) section in the [Program - Settings section of the PageGate Admin](#).
- The PageGate server must be running. If you're running PageGate as an application, launch the PageGate server from the shortcut in the start menu. If you're running PageGate as a Windows Service, start the PageGate service. For more information, [see the Run Styles](#) section of the documentation.

PageGate Log

This log is a record of all protocol negotiations handled by this connector.



Button	Function
Refresh	Displays the most current entries in the log.
Print	Sends a copy of everything listed in the log to a printer.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.

Carriers

Carriers are, generally speaking, the service providers for the devices you want to send messages to, whether you're sending to cell phones or pagers. When the term is used in PageGate, however, it's referencing one of the outbound communication methods the program supports.

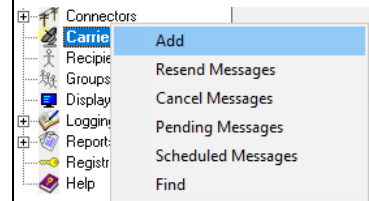
In concept, a carrier tells PageGate how to talk to the outside world with one of the supported protocols (see below), a [recipient](#) tells PageGate who to talk to and the [connector](#) modules do the talking.

PageGate Admin: How to Configure PageGate

6

Right-clicking on Carriers will provide a context menu with four options.

- [Add](#)
This option allows you to create a new carrier.
- [Requeue Messages](#)
This option allows you to requeue a range of previously delivered messages to ALL carriers.
- [Cancel Messages](#)
This option allows you to cancel all past due, pending and/or repeating messages for ALL carriers.
- [Pending Messages](#)
This option will display a list of all messages for all carriers in the pending queue.
- [Find](#)
This option allows you to search the list of carrier names.



PageGate supports the following delivery protocols:

- [AWS SNS](#) (Amazon Web Services Simple Notification System)
A method used to communicate with the AWS SNS system for SMS delivery.
- [SMTP](#) (Simple Mail Transfer Protocol)
The protocol used to deliver email.
- [SNPP](#) (Simple Network Paging Protocol)
A standard message delivery protocol for internet communication to a wireless provider.
- [WCTP](#) (Wireless Communications Transfer Protocol)
A standard message delivery for Internet communications to a wireless provider (both HTTP and HTTPS are supported).
- [GSM-AT](#) (Global System for Mobile communication AT Command Set)
A method used to communicate with a cellular modem or other connected wireless device that provides a COM port in Windows and supports the AT command set for SMS delivery.
- [GSM-AT-IP](#) (Global System for Mobile communication AT Command Set over IP)
A method used to communicate with a cellular modem, router, gateway or other connected wireless device that accepts AT commands by TCP for SMS delivery.
- [HTTPGET](#)

PageGate Admin: How to Configure PageGate

The GET method of passing CGI parameters to a web API.

- [HTTPPOST](#)

The POST method of passing CGI parameters to a web API.

- [File](#)

Output a message as an ascii formatted file to a specified directory.

- [XMPP](#) (Extensible Messaging and Presence Protocol)

A method used to communicate with a variety of notification and chat programs, such as Jabber.

- [TAP](#) (Telocator Alphanumeric Protocol)

The standard message delivery protocol for analog modem, or direct serial communications to paging providers and terminals.

- [TAP-IP](#) (Telocator Alphanumeric Protocol delivered by TCP/IP)

This protocol is used to send messages to a receiving TAP system, normally paging transmitters, by TCP/IP.

- [FAX](#) (Facsimile Machines)

Message delivery to destination Fax machines via faxmodem.

- [FAXSrv](#)

This protocol allows PageGate to integrate with a Microsoft fax server for facsimile messages.

- [RawSerial](#)

Transmission of message text over a standard serial/RS232 port (direct connect or via modem).

- [RawTCP](#)

Transmission of message text to a host via a TCP/IP connection

- [TONE](#) (Dual Tone Multi-Frequency)

A way to deliver touch tone messages to older numeric-only pagers and similar equipment.

- [UCP](#) (Universal Computer Protocol)

A delivery protocol for analog modem, or direct serial communications to a wireless provider.

For a detailed and contextualized breakdown of what these protocols mean for delivering your messages, have a look at the [How PageGate Sends Messages](#) section of the documentation.

PageGate Admin: How to Configure

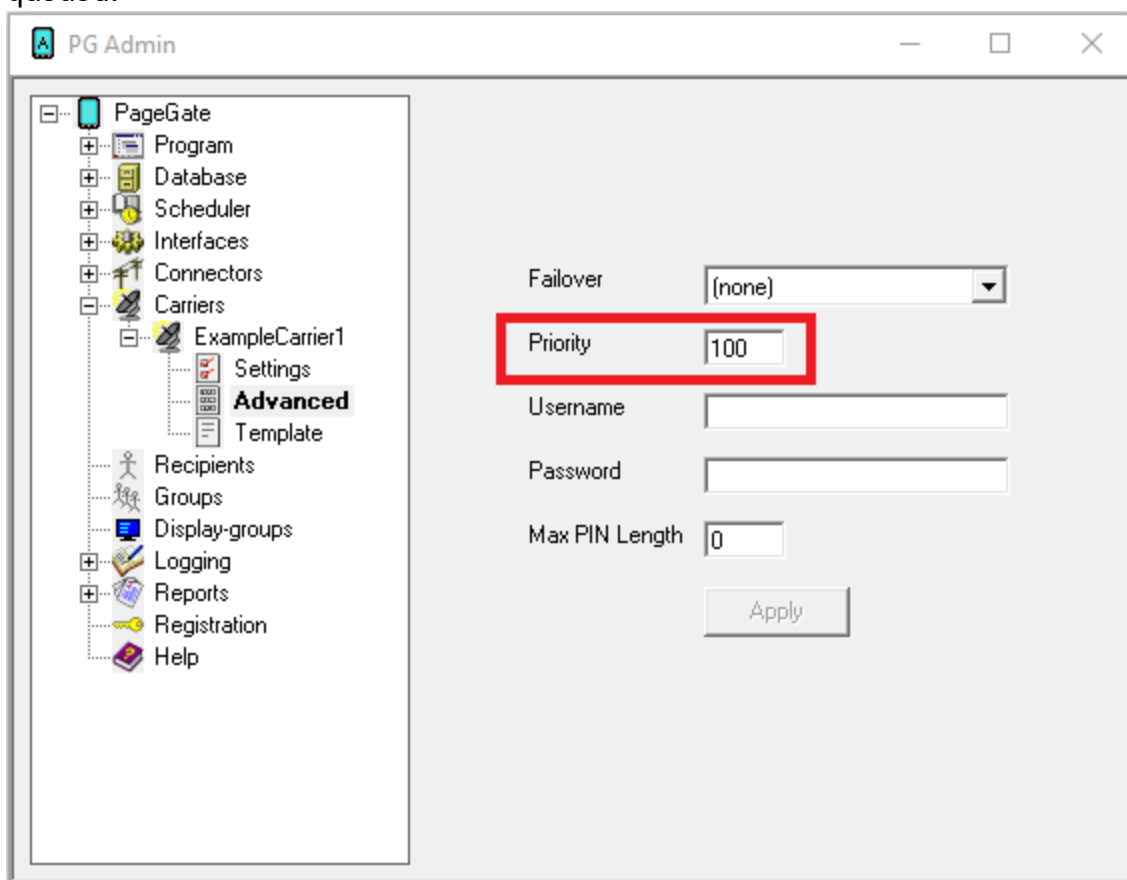
PageGate Prioritizing Messages

PageGate allows you to prioritize your traffic by setting a priority value at both a carrier and recipient level and the 'Priority' field can be found in the Advanced sub-section of every carrier and recipient.

The scale of priority is 1 (the most important) to 32000 (the least important). By default, all carriers and recipients start with a priority value of 100.

Entries that share a priority value are treated equally. Entries with lower values are treated as more important, entries with higher values are treated as less important.

Carrier level priority determines the order in which carriers should have their messages queued.

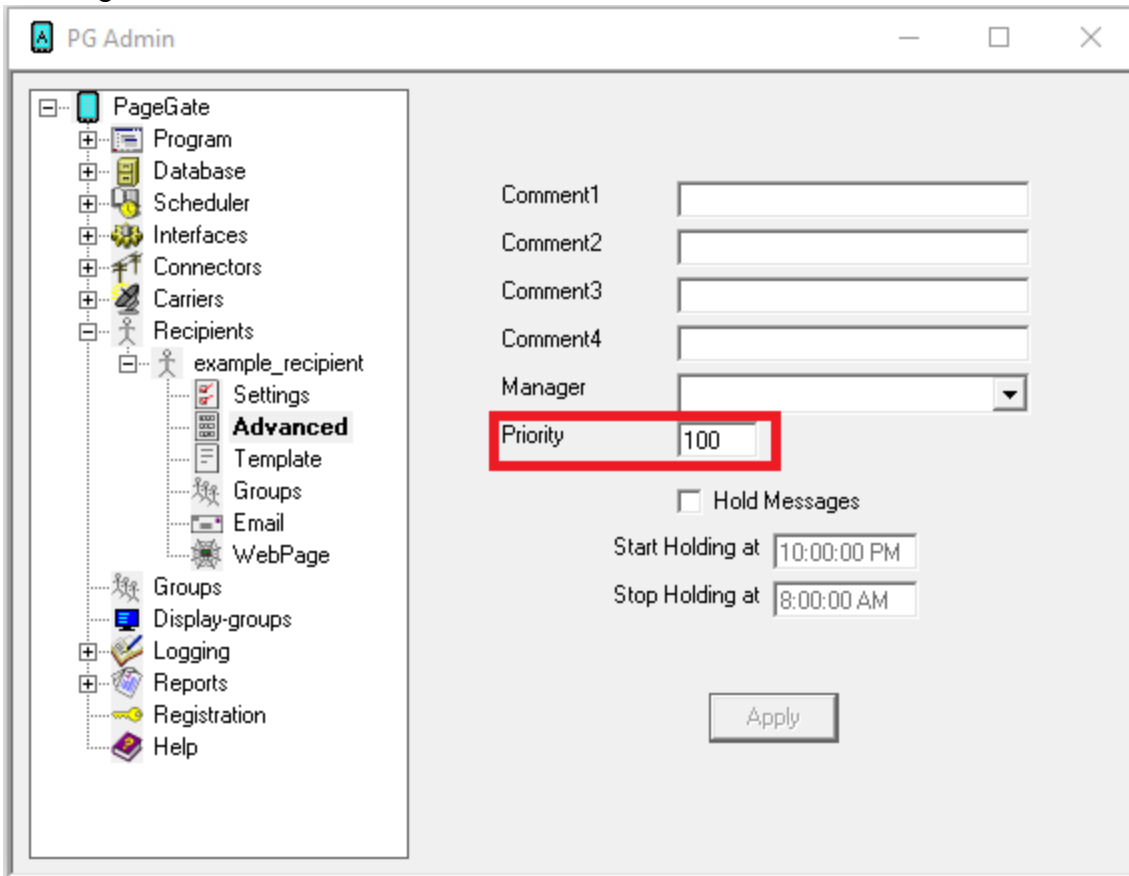


For example, let's say that you use I Am Responding and Active911's message aggregation service and that you also deliver regular email, SMS and pages through PageGate. If you set the priority value for I Am Responding and Active911 to something lower than the priority value for the other carriers, PageGate will always schedule the I Am Responding and Active911 messages to go out before anything else.

Carrier priority values are determined in a carrier's [Advanced sub-section](#).

PageGate

Recipient level priority determines the order in which recipients should receive their messages.



For example, let's say that you had a list of 100 recipients and that we had 10 recipients who needed to receive their traffic before anyone else. If you set the priority value for those recipients to something lower than the priority value for the other recipients, PageGate will always schedule their messages first. To give a slightly more practical example, let's say that you have a dispatch supervisor or fire marshal or IT director that needed to be notified before anyone else in the system. You could edit the value of that recipient's Priority field to be lower than the other recipients and that would ensure that they always received their messages first.

Recipient priority values are determined in a recipient's [Advanced sub-section](#).

Creating a new Carrier

To create a new carrier, you'll need some basic information about the types of messages you want to deliver and we would strongly recommend having a look at the [How PageGate Sends Messages](#) section of the documentation for more information on that. However, once you know which protocol(s) you want to implement for message delivery, you'll need to create a new carrier or series of carriers.

- [How to create an AWS SNS Carrier](#) (Amazon Web Services Simple Notification System)

PageGate Admin: How to Configure PageGate

A method used to communicate with the AWS SNS system for SMS delivery.

- [How to create an SMTP Carrier](#) (Simple Mail Transfer Protocol)
The protocol used to deliver email and email to SMS messages.
- [How to create an SNPP Carrier](#) (Simple Network Paging Protocol)
A standard message delivery protocol for internet communication to a wireless provider.
- [How to create a WCTP Carrier](#) (Wireless Communications Transfer Protocol)
A standard message delivery for Internet communications to a wireless provider (both HTTP and HTTPS are supported).
- [How to create a GSM-AT Carrier](#) (Global System for Mobile communication AT Command Set)
A method used to communicate [cellular modem](#) or [other connected wireless device](#) that provides a COM port in Windows and supports the AT command set for SMS delivery.
- [How to create a GSM-AT-IP Carrier](#) (Global System for Mobile communication AT Command Set over IP)
A method used to communicate with a with a NotePage SMS Gateway App enabled Android phone, [cellular modem](#), [router](#), [gateway](#) or [other connected wireless device](#) that accepts AT commands by TCP for SMS delivery.
- [How to create an HTTPGET Carrier](#)
The GET method of passing parameters to a web API.
- [How to create an HTTPPOST Carrier](#)
The POST method of passing parameters to a weborm, Web API or REST API.
- [How to create a File Carrier](#)
Output a message as an ascii formatted file to a specified directory.
- [How to create an XMPP Carrier](#) (Extensible Messaging and Presence Protocol)
A method used to communicate with a variety of notification and chat programs, such as Jabber.
- [How to create a TAP Carrier](#) (Telocator Alphanumeric Protocol)
This protocol is used to send messages to a receiving TAP system, normally paging transmitters, by dial-up modem or serial connection.
- [How to create a TAP-IP Carrier](#) (Telocator Alphanumeric Protocol delivered by TCP/IP)
This protocol is used to send messages to a receiving TAP system, normally paging transmitters, by TCP/IP.

PageGate Admin: How to Configure

PageGate

- [How to create a FAX carrier](#) (Facsimile Machines)

Message delivery to destination Fax machines via faxmodem.

- [How to create a FAXSrv Carrier](#)

This protocol allows PageGate to integrate with a Microsoft fax server for facsimile messages.

- [How to create a RawSerial Carrier](#)

Transmission of message text over a standard serial/RS232 port (direct connect or via modem).

- [How to create a RawTCP Carrier](#)

Transmission of message text to a host via a TCP/IP connection

- [How to create a Touch-Tone/DTMF Carrier](#) (Dual Tone Multi-Frequency)

A way to deliver touch tone messages to older numeric-only pagers and similar equipment.

- [How to create a UCP Carrier](#) (Universal Computer Protocol)

A delivery protocol for analog modem, or direct serial communications to a wireless provider.

Creating an AWS SNS Carrier

Step 1: Create an Amazon Web Services Account

- 1) Go to <https://aws.amazon.com/>
- 2) Click an AWS Create Account
- 3) Enter the email address and password.
- 4) Enter payment information (required by Amazon)
- 5) Choose your support plan.

Step 2: Create an Access Key for PageGate to use

Creating an access key can be done by providing PageGate access to the root account and password or you can tie the AWS SNS permissions to a specific user account..

If you want PageGate to use the root account, you would do the following:

- 1) Log in to your AWS account.
- 2) In the upper right hand corner, click on your account's name and select 'My Security Credentials'
- 3) Click 'Create new Access Key'

PageGate

- 4) Copy the 'Access Key ID' and 'Secret Access Key' values as we'll need both in PageGate.

If you want to grant the ability to use AWS SNS to an existing account, you would do the following:

- 1) Log in to your AWS account.
- 2) In the upper right hand corner, click on your account's name and select 'My Security Credentials'
- 3) On the left hand side, click 'Users'
- 4) Select the user in question.
- 5) Click 'Add Permissions'
- 6) Select 'Attach existing policies directly'
- 7) In the 'Filter policies' field, enter: SNS
- 8) Check 'AmazonSNSFullAccess'
- 9) Click Next.
- 10) Click 'Add Permissions'
- 11) Select the 'Security Credentials' tab for that user.
- 12) Click 'Create Access Key'
- 13) Copy the 'Access Key ID' and 'Secret Access Key' values as we'll need both in PageGate.

Step 3: Configure AWS SNS in PageGate

- 1) Run PG Admin.
- 2) Right click Carriers, select Add
- 3) Specify the following:

Carrier: AWSSNS

Protocol: AwsSns

Host: [Please see Amazon's SNS documentation](#)

<https://docs.aws.amazon.com/sns/latest/dg/sns-supported-regions-countries.html>

The 'Endpoint' value is what you will use in the this field.

PageGate Admin: How to Configure

PageGate

Examples.

sns.us-east-1.amazonaws.com

sns.us-west-2.amazonaws.com

Max Chars: 160

- 4) Click Apply to add the new delivery method.
- 5) Go to Carriers - AWSSNS - Advanced
- 6) Enter the Access Key ID value in the Username field.
- 7) Enter the Secret Access Key value in the Password field.
- 8) Click Apply.

Step 4: Create Recipients tied to the AWSSNS method of delivery

- 1) Right click Recipients, select Add.
- 2) Specify the following:

Full Name: The name of the recipient

Recipient: This field will automatically populate but can be edited and may ONLY contain alphanumeric characters, underscores and dashes.

Type: Normal

Carrier: AWSSNS

Failover: See [Failover Scenarios](#)

ID/PIN: cell phone number, including country code.

NOTE: For messaging in North America, it is imperative that you provide the 1 in front of the phone number. AWS SNS treats the first character of the phone number as the country code and you can unintentionally send international SMS if you do not provide the 1.

<u>Correct</u>	<u>Incorrect</u>
17818290500	781829500

Max Chars: Specify the number of characters you want the ability to send to this recipient. If this value is higher than the carrier's "Max Chars" value, PageGate will break up long messages in to multiple blocks.

- 3) Click Apply.

PageGate Creating an SMTP Carrier

SMTP is the protocol used to send email of all varieties, including those delivered as SMS; nearly every US and Canadian carrier supports text messaging to their phones using a public access SMTP/Email gateway.

PageGate has a full mail server engine as a core part of the program and this gives you the ability to send SMTP/Email messages in one of two ways: Direct and Relayed

To explain the difference between the two, direct delivery allows PageGate to use its own mail server engine to directly negotiate with the receiving mail servers. Relayed delivery configures PageGate to pass all of its traffic through a verified SMTP server for delivery, like an email client.

Technically, direct delivery is more efficient and reliable because it removes at least one hop in the delivery sequence and also means that PageGate isn't reliant on another mail server for delivery. However, when you have PageGate use direct delivery for SMTP/Email, it has to abide by all of the rules required of a mail server because the program will be functioning as a mail server to deliver SMTP/Email on behalf of your email domain. This is incredibly important to note because, in an effort to combat spam, all major US and Canadian carriers pay respect to SPF records.

SPF stands for "Sender Policy Framework" and is a record of public IP addresses that are allowed to deliver email on behalf of your email domain. This record is published by whoever owns and operates the email domain. If the PageGate server's public IP address is not part of your email domain's SPF record, the major carriers will filter your messages as potential spam for failing the SPF validation check. If you continue to send traffic that fails SPF validation checks, the receiving servers will eventually blacklist your site from sending any traffic to them.

Due to some of the complications involved in configuring PageGate for direct delivery, it is usually more practical to configure PageGate to relay its SMTP/Email traffic through your company or ISP's SMTP/Email server. However, doing so makes PageGate reliant on that mail server for message delivery.

For information on how to configure a "Relayed Delivery" SMTP carrier, [click here](#).

For information on how to configure a "Direct Delivery" SMTP carrier, [click here](#).

Creating a Relayed SMTP Carrier

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the SMTP protocol.
- 4) Enter a name in the Carrier field.

PageGate Admin: How to Configure PageGate

6

5) In the Carrier Domain field, enter the messaging domain of this carrier. [Click here for a list of known SMTP messaging domains.](#)

Note: Do not use the @ in this field. For example, you would use vtext.com as the Carrier Domain if you were configuring Verizon and you would use txt.att.net if you were configuring AT&T.

6) In the Port field, specify the port and handshaking method required by your SMTP/Email server.

7) In the Max Chars field, enter the maximum number of characters you want to send in a single message. For SMS hosts, set this value to 160. For pager hosts, set this value to 240. For standard email, set this value to 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

8) Click Apply.

9) Go in to the Advanced sub-section of your new carrier.

10) Select Deliver through outgoing mail server.

11) Enter your SMTP server address in the Relay Server field.

12) If your SMTP server requires authentication (a username and password), enter your authentication credentials in the username and password fields.

13) Click Apply.

14) Go to Program - Template.

15) In the Settings section, set the InternetDomain = variable to match the email domain of the SMTP server you're relaying your messages through.

16) Click Apply.

[Click here for a list of Settings Variables that can be used with SMTP Carriers](#)

PageGate

Creating a Direct Delivery SMTP Carrier

When you have PageGate use direct delivery for SMTP/Email, it has to abide by all of the rules required of a mail server because the program **will be functioning as a mail server** to deliver SMTP/Email on behalf of the email domain you specify in the program's [global template](#). This is incredibly important to note because, in an effort to combat spam, all major US and Canadian carriers pay respect to SPF records.

SPF stands for "Sender Policy Framework" and is a record of public IP addresses that are allowed to deliver email on behalf of your email domain. This record is published by whoever owns and operates the email domain. If the PageGate server's public IP address is not part of your email domain's SPF record, the major carriers will filter your messages as potential spam for failing the SPF validation check. If you continue to send traffic that fails SPF validation checks, the receiving servers will eventually blacklist your site from sending any traffic to them.

The public IP address that the PageGate server sits behind must be a part of the SPF record for your email domain.

To use direct delivery, port 25 must be accessible from the PageGate server. If port 25 is not open, you cannot use direct delivery and must use [relayed delivery](#) instead.

To determine if port 25 is open for the PageGate server, there is a test you can run with the Telnet Client in Windows or with a program like Putty or Hyperterminal.

After installing the Telnet Client in Windows, [click here](#) for the steps to test if port 25 is open.

If port 25 is open AND the public IP address of the PageGate server is part of the SPF record for your email domain, you can configure PageGate to directly deliver SMTP/Email traffic.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the SMTP protocol.
- 4) Enter a name in the Carrier field.
- 5) In the Carrier Domain field, enter the messaging domain of this carrier. [Click here for a list of known SMTP messaging domains.](#)

Note: Do not use the @ in this field. For example, you would use vtext.com as the Carrier Domain if you were configuring Verizon and you would use txt.att.net if you were configuring AT&T.

- 6) In the Port field, select: 25

PageGate Admin: How to Configure PageGate

6

- 7) In the Max Chars field, enter the maximum number of characters you want to send in a single message. For SMS hosts, set this value to 160. For pager hosts, set this value to 240. For standard email, set this value to 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 8) Click Apply.
- 9) Go to Program - Template.
- 10) In the Settings section, set the InternetDomain = variable to match the email domain whose SPF record the PageGate system's public IP address is a part of.
- 11) Click Apply.

[Click here for a list of settings Template Variables that can be used with SMTP Carriers](#)

Testing Port 25

To check if port 25 is open for your PageGate server, you'll first need to install the telnet client. After the telnet client has been installed, this test will let you know if port 25 is open or closed.

- 1) In Windows, run a Command Prompt.
- 2) Type the following and hit enter: nslookup
- 3) Type the following and hit enter: set type=mx
- 4) Type the following and hit enter: vtext.com
- 5) Make a note of the first mail exchanger response.
- 6) Type the following and hit enter: exit
- 7) Type the following and hit enter: telnet <First mail exchanger> 25
In this case, we would use telnet vrz-sms.mx.a.cloudfilter.net 25

If port 25 is open and there are no outstanding issues with the public IP address the system sits behind, you'll receive a 220 response and can use direct delivery.

PageGate

If port 25 is open but there is an outstanding issue with the public IP the system sits behind, you'll receive a 554 or a 421 rejection notice. These responses indicate that your public IP is on a blacklist and you will need to go through the process of IP remediation.

This process can vary widely depending on what blacklist(s) your IP may be on, so the best thing would be to go to each respective blacklist's site to go through their remediation process.

If port 25 isn't open at all, you'll receive a response that says "Unable to establish a connection to the host on port 25". If port 25 isn't open, you may be able to have it opened but you will need to speak with your network administrator or internet service provider's tech support.

If port 25 is closed and cannot be opened, you must [relay your traffic through an SMTP server](#).

Creating an SNPP Carrier

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the SNPP protocol.
- 4) Enter a name in the Carrier field.
- 5) Enter the SNPP host you want to communicate with.
- 6) Unless the host specifically requires a connection on something other than port 444, leave the port at 444.
- 7) In the Max Chars field, enter the maximum number of characters you want to send in a single message. For SMS hosts, set this value to 160. For pager hosts, set this value to 240. The maximum value for this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 8) Click Apply.

If this host requires authentication (a username and password), you can configure that in the Advanced sub-section of the carrier. Please see the [Editing Carriers](#) portion of the manual.

PageGate Admin: How to Configure PageGate

6

[Click here for a list of Settings Variables that can be used with SNPP Carriers](#)

Creating a WCTP Carrier

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the WCTP protocol.
- 4) Enter a name in the Carrier field.
- 5) Enter the WCTP host you want to communicate with.
Note: Do not use the HTTP or HTTPS prefix in the Host field. These values are automatically inserted based on port selection.
- 6) If the host requires a connection on a secure port, select port 443. Otherwise, leave the port at 80.
- 7) In the Max Chars field, enter the maximum number of characters you want to send in a single message. For SMS hosts, set this value to 160. For pager hosts, set this value to 240. The maximum value for this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 8) Click Apply.

If this host requires authentication (a username and password), you can configure that in the Advanced sub-section of the carrier. Please see the [Editing Carriers](#) portion of the manual.

[Click here for a list of Settings Variables that can be used with WCTP Carriers](#)

Creating a GSM-AT Carrier

Cellular Modems

The GSM-AT protocol is used when you want to communicate with a cellular modem by COM port, whether physical or virtual.

PageGate Admin: How to Configure

6

PageGate

A cellular modem connects to a PC by USB or RS232 and provides a COM port in the operating system.

If PageGate is running in a virtual environment, you can also connect a cellular modem to a COM port virtualizer that provides the COM port to the virtual machine. This grants PageGate access to the cellular modem, which allows it to send and receive text messages just like a cell phone does.

Good examples of cellular modem are:

- Multitech's MTC-LNA4
- Microhard's IP4NG
- AirPrime 73xx and 75xx
- Skyus-DS
- BlueOcean GSM and GPRS series
- Duxbury 120 and 180
- Siemens MTC45MC, MTC45TC, TC35, TC65
- Wavecom Fastrack GSM Modems

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the GSM-AT protocol.
- 4) Enter a name in the Carrier field.
- 5) Set the Baud Rate, Parity, Data Bits and Stop Bits to match the device's requirements.

By default, most cellular modems use the following parameters:

Baud Rate: 115200

Parity: None

Data Bits: 8

Stop Bits: 1

- 6) The Init String field optional for a GSM-AT carrier. To have the program perform a signal quality check on the cellular modem before each delivery, click inside the Init String field and type the following: AT+CSQ
- 7) Set the Max Chars to: 160

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character

PageGate Admin: How to Configure PageGate

6

limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

8) Click Apply.

9) Go to Connectors - Connector X - Settings.

Note: Any connector can use a cellular modem but a connector can only be tied to one cellular modem at a time. Multiple connectors cannot reference the same cellular modem.

10) Set the Serial Port field to match the COM port of your modem.

11) Click Apply.

When configuring PageGate to use a cellular router or gateway, each manufacturer uses a slightly different command set to perform the same tasks and there are [certain Settings template variables](#) that you can apply to GSM-AT-IP carriers to decrease the amount of time it takes to negotiate connections with your hardware.

To implement a two-way messaging system, please see the [Two Way Messaging](#) section of the documentation.

[Click here for a list of Settings Variables that can be used with GSM-AT Carriers](#)

Creating a GSM-AT-IP Carrier

Cellular Gateways/Routers

The GSM-AT-IP protocol is used when you want to communicate with a cellular modem, router or gateway by Telnet or SSH.

Cellular routers and gateways can be accessed via TCP by Telnet or SSH. You don't have to connect these devices to a PC, they just need to be connected to your network and configured appropriately to receive messaging commands.

Good examples of cellular routers/gateways are:

- Multitech's MTR-LNA7
- Airlink's GX450, RV50, RV55 and LX40
- Microhard IP4nGii
- Cisco Series 800, 900

PageGate

- Cisco AER1600/1650 Series, AER31x0, AER2100, AER2200 Series, COR IBR200, COR IBR600B/C Series, COR IBR900 Series, COR IBR1100 Series, COR IBR1700, MBR1400, MBR1200B, ARC CBA850, ARC CBA750B*, IBR300, IBR6x0, IBR11x0

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the GSM-AT-IP protocol.
- 4) Enter a name in the Carrier field.
- 5) In the Host field, enter the host IP address of the cellular router or gateway.
- 6) Edit the Port field to match the port required by the cellular router or gateway.
- 7) The Init String field optional for GSM-AT-IP carriers. To have the program perform a signal quality check on the cellular modem before each delivery, click inside the Init String field and type the following: AT+CSQ
- 8) Set the Max Chars to: 160

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 9) Click Apply.

When configuring PageGate to use a cellular router or gateway, each manufacturer uses a slightly different command set to perform the same tasks and there are [certain Settings template variables](#) that you can apply to GSM-AT-IP carriers to decrease the amount of time it takes to negotiate connections with your hardware.

To implement a two-way messaging system, please see the [Two Way Messaging](#) section of the documentation.

[Click here for a list of Settings Variables that can be used with GSM-AT-IP Carriers](#)

PageGate supports the ability to use HTTP GET and POST methods which can be used to submit web forms, request information from websites and/or WEB APIs and interact with REST APIs.

The website, web form or API you interact with will directly determine what you need to configure in PageGate.

GET and POST are conceptually similar to each other but perform their functions in completely different ways.

The GET method carries parameters in the URL string and the POST method carries parameters in the request body.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the HTTPGET protocol.
- 4) Enter a name in the Carrier field.
- 5) Set the host field to the URL of the API you wish to reach.
- 6) Set the Port field to the port required by the web API.
- 7) Set the Max Chars to 160 for SMS or 240 for pager messages. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 8) Click Apply.
- 9) Go in to the Template sub-section of your new HTTPGET carrier.
- 10) [Configure a template for this carrier.](#)

Templates for HTTP GET Carriers

Templates for HTTP GET carriers tell PageGate what variables and data to declare in the URL passed to the API.

All variables are declared in the 'Text' section of the template as follows:

PageGate Admin: How to Configure

6

PageGate

|<name1>|<value1>|
|<name2>|<value2>|
etc.

Each value you specify will need to correspond to a variable that the receiving API will recognize. For example, let's say we wanted to send a message to an API that looked for to, from, subject and text as variables. We would need to format the text template as follows:

to	%RecipientId%
from	%Sender%
subject	%Subject%
text	%Message%

NOTE: | a pipe character.

The following parameters can be specified in the 'Settings' section of an HTTP GET carrier

Variable	Possible Values	Example Statement	Description
Response Recipient	Recipient Group	Response Recipient = A recipient or group's name as it exists in the PG Admin	When the HTTP GET transaction completes, the specified recipient or group will receive any responses from the receiving API/web host. For more information, please see the Two Way Messaging section of the documentation.
InterMessageDelay	Milliseconds	InterMessageDelay = 1000	This variable forces PageGate to delay for the specified number of milliseconds (1000 = 1 second) after delivering each message.
From	Text	From = CAD	This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.
To	Text	To = 7818290500	This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.

Creating an HTTP POST Carrier

This protocol allows PageGate to communicate with a webform, REST and Web APIs.

Since the API you interact with will likely require authentication, it's important to note there are multiple styles of authentication and the APIs you use will directly determine which to implement in PageGate.

Using the Username and Password in the carrier settings configure PageGate to use Basic HTTP Authentication, which is fairly standard.

Other forms of authentication, like authorization headers or required API keys in the XML/JSON/etc body, are configured in the [POST carrier's template](#).

For example, another standard is to use the Authorization header which is typically in the format:

Authorization: <type> <credentials>

<type> and <credentials> will be defined in the receiving APIs documentation.

Some APIs use a custom header or authentication included in HTTP GET parameters or POST parameters.

Authorization can also be part of the POST body. For example, you might have to include API keys or something similar as part of the JSON or XML body.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the HTTPPOST protocol.
- 4) Enter a name in the Carrier field.
- 5) Set the host field to the URL of the API you wish to reach.
- 6) Set the Port field to the port required by the API or webform, typically 443.
- 7) Set the Max Chars to 160 for SMS or the maximum number of characters the API can handle in a single transmission. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars

PageGate Admin: How to Configure

6

PageGate

and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 8) Click Apply.
- 9) Go in to the Template sub-section of your new HTTPPOST carrier.
- 10) [Configure a template for this carrier.](#)

Templates for HTTP POST Carriers

NOTE: When referencing any non-printable character, such as a line feed, carriage return, horizontal tab or similar, you must use the ASCII code for that character and it must be prefixed by a backslash (\).

Templates for HTTP POST carriers tell PageGate how to interact with and structure data passed to a web form, Web API or REST API.

It's important to gather as much information as you can about what API or webform you're interacting with and their documentation will typically provide required variables and how they need to be implemented.

Conceptually, the 'Settings' section of a POST carrier template tells PageGate if the API requires any special interaction and the 'Text' section determines what variables to declare and how to structure the message body.

These values can be used in the 'Settings' section of an HTTP POST carrier:

Variable	Possible Values	Example Statement	Description
Response Recipient	Recipient Group	ResponseRecipient = A recipient or group's name as it exists in the PG Admin	When the HTTP GET or POST transaction completes, the specified recipient or group will receive any responses from the receiving API/web host. For more information, please see the Two Way Messaging section of the documentation.

HttpPostType	Head Body or UrlHeadBody	<p>HttpPostType = HeadBody</p> <p>HttpPostType = UrlHeadBody</p>	<p>HttpPostType = HeadBody configures an HTTP POST carrier to provide header information followed by structured data, usually JSON or XML, in the body.</p> <p>HttpPostType = UrlHeadBody configures an HTTP POST carrier to pass parameters in the URL, more like the GET method. This is not a typical use of HTTP POST but is used/required by some APIs.</p>
HttpSuccessResponse	Text	See Description	<p>This variable tells PageGate to scan any responses back from a GET or POST request for the required keywords or keyphrases. If those keywords or keyphrases are found, the message is treated as successfully delivered.</p> <p>This variable is both case and punctuation sensitive and PageGate will look for an exact match to what you enter.</p> <p>Multiple search terms may be used and you'll need to separate each item with a pipe.</p> <p>For example, this would only scan for the word success: <code>HttpSuccessResponse = success</code></p> <p>This would scan for the word success in the <Reply> tag: <code>HttpSuccessResponse = <Reply>success</Reply></code></p> <p>This would scan for the word success in the <Reply> tag OR the word hijinks: <code>HttpSuccessResponse = <Reply>success</Reply> hijinks</code></p> <p>This would scan for the word success OR the word hijinks OR the word actualization: <code>HttpSuccessResponse = success hijinks actualization</code></p> <p>If this variable is included, any response to a GET/POST request that doesn't include the search term(s) will be treated as a failure.</p>
HttpErrorResponse	Text	From = CAD	<p>This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable,</p>

PageGate Admin: How to Configure PageGate

6

			you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.
From	Text	From = CAD	This variable overrides the value passed to PageGate by the input modules/APIs. By specifying this variable, you're telling PageGate to use a single, standardized sender's name for all traffic controlled by this template.
To	Text	To = 78182905 00	This variable overrides the ID/PIN or Email values. By specifying this variable, you're telling PageGate to send all traffic processed by this template to the specified ID or Email address instead.

This is how to use the 'Text' section of an HTTP POST carrier:

Each value you specify will need to correspond to a variable that the receiving API will recognize. For example, let's say we wanted to send a message using [Twilio's API](#). We know they're looking that looked for to, from and body as variables. We would need to format the text template as follows:

```
|to|{%RecipientId%|  
|from|{%Sender%|  
|body|{%Message%|
```

NOTE: | is a pipe character and is found on most keyboards above the backslash character, \.

Some APIs, like [MessageBird](#), require an authentication variable in the body of the POST and those are declared like any other variable.

Example:

```
|Authorization|AccessKey PasteYourAccessKeyHere|
```

```
|recipients|{%RecipientId%|  
|originator|{%Sender%|  
|body|{%Message%|
```

Creating a File Carrier

This type of carrier tells PageGate to write a file to a specified directory when a message needs to be sent. This is often used in conjunction with PageGate's [two-way messaging system](#) but can be implemented any time you need messages written out as a file to a specified location.

It is also important to note that you must apply a Text template to any File protocol carrier that you create.

PageGate Admin: How to Configure PageGate

6

This protocol also allows you to write files for the [GetASCII API](#) to process, which allows you to take input from other APIs, whether they're PageGate's or a web API, and process it in to a readable message.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the File protocol.
- 4) Enter a name in the Carrier field.
- 5) In the File Path field, enter the location to which you want these files output. This can be set to a local path (example: c:\PageGateData\ASCII\ or it can be set to a network path (\servername\share\). If this carrier is referencing a network path, the PageGate Connector services will need a service account to grant them access to the necessary network resource. [Click here](#) for more information.
- 6) Select the extension you want the files to have.
- 7) Set the Max Chars field to the maximum number of characters you want output in a single file. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 8) Click Apply.

Now that we have the carrier created, go in to its Template sub-section.

In the Text section of the template, you'll want to configure a string of [Macros](#) to tell PageGate what should be output. For example, if we used the following:

```
%Recipient%  
%Sender%  
%Message%
```

PageGate

That tells PageGate to output the recipient's name followed by a carriage return, line feed then the recorded sender value followed by a carriage return, line feed then the body of the message in the file written.

As another example, if we used the following:

```
%Message% has failed to %Recipient% on %Date% at %Time%
```

That tells PageGate to output the body of the message has failed to the recipient in question on the date of its occurrence at the time of its occurrence.

This allows you to implement the File protocol in any number of ways, from feeding information back in to PageGate's [GetASCII API](#) and [Filter Pack](#) to outputting a copy of any message as a file for safe keeping or secondary processing by other applications.

Also, bear in mind that while these are examples of what can be done with this type of carrier, it's important to note the base action in play:

You're having PageGate output a text file to a specified location.

Creating an XMPP Carrier

This protocol allows you to communicate with XMPP servers, which are predominantly used for chat messages. This is a one-way implementation of the protocol and PageGate does not currently listen for replies from the XMPP server.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the XMPP protocol.
- 4) Enter a name in the Carrier field.
- 5) Enter the XMPP host you want to communicate with.
- 6) Click inside the Port field and enter the port on which the XMPP server requires a connection.
- 7) In the Max Chars field, enter the maximum number of characters you want to send in a single message. The maximum value for this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

PageGate Admin: How to Configure

PageGate

8) Click Apply.

If this host requires authentication (a username and password), you can configure that in the Advanced sub-section of the carrier. Please see the [Editing Carriers](#) section of the manual.

Creating a TAP Carrier

The TAP protocol is used to communicate with paging systems and any other system that supports the TAP protocol and can be implemented in any or all of three ways:

- [Serial](#)

This method configures PageGate to use a serial/RS232 connection to a paging transmitter.

- [Modem](#)

This method configures PageGate to use a dial-up modem to dial in to the receiving paging transmitter to deliver messages.

Creating a Serial/RS232 TAP Carrier

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the TAP protocol.
- 4) Enter a name in the Carrier field.
- 5) Leave the Phone Number field empty.
- 6) Click inside the Init String field and type the following: direct
- 7) Set the Baud Rate, Parity, Data Bits and Stop Bits to match the receiving TAP system's requirements.

By default, most TAP systems use:

Baud Rate: 1200

Parity: Even

Data Bits: 7

Stop Bits: 1

- 8) Set the Max Chars value to: 240

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the

PageGate Admin: How to Configure

6

PageGate

message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

9) Click Apply.

10) Go to Connectors - Connector X - Settings.

Note: Any connector can use a serial connection but a connector can only be tied to one serial connection at a time. Multiple connectors cannot reference the same serial port.

11) Set the Serial Port field to match the serial port the RS232 cable is plugged in to.

12) Check Direct Connect.

13) Click Apply.

[Click here for a list of Settings Variables that can be used with TAP Carriers](#)

Creating a Modem TAP Carrier

1) Right click Carriers.

2) Select Add.

3) Select the TAP protocol.

4) Enter a name in the Carrier field.

5) Enter the phone number of the receiving TAP system in the Phone Number field.

6) Set the Baud Rate, Parity, Data Bits and Stop Bits to match the receiving TAP system's requirements.

By default, most TAP systems use:

Baud Rate: 1200

Parity: Even

Data Bits: 7

Stop Bits: 1

7) Click inside the Init String field and type the following: direct

8) Set the Max Chars value to 160 for SMS or 240 for pagers. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the

PageGate

message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

9) Click Apply.

10) Go to Connectors - Connector X - Settings.

Note: Any connector can use a modem but a connector can only be tied to one modem at a time. Multiple connectors cannot reference the same modem.

11) Set the Serial Port field to match the COM port of the modem.

12) Click Apply.

[Click here for a list of Settings Variables that can be used with TAP Carriers](#)

Creating a TAP-IP Carrier

1) Right click Carriers.

2) Select Add.

3) Select the TAP-IP protocol.

4) Enter a name in the Carrier field.

5) Set the Host field to the IP of the TAP receiver.

6) Set the Port field to the port required by the TAP receiver.

7) Set the Max Chars value to: 240

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

8) Click Apply.

[Click here for a list of Settings Variables that can be used with TAP Carriers](#)

Creating a FAX Carrier

This protocol allows you to tell PageGate to send a fax using a fax capable dial-up modem.

NOTE: While not absolutely required, we would recommend creating a [Faxing Template](#) for any FAX Carriers you create.

PageGate Admin: How to Configure PageGate

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the FAX protocol.
- 4) Enter a name in the Carrier field.
- 5) Set the Init String field to: (auto)
- 6) Set the Fax Class required by the receiving fax machine(s).
- 7) Set the Max Chars value to: 6000
- 8) Click Apply.
- 9) Go to Connectors - Connector X - Settings.

Note: Any connector can use a modem but a connector can only be tied to one modem at a time. Multiple connectors cannot reference the same modem.

- 10) Set the Serial Port field to match the COM port of your modem.
- 11) Click Apply.

Creating a FAXSrv Carrier

NOTE: This protocol will only work with a Microsoft Fax Server.

For more information on how to configure a Microsoft Fax Server, please review the Microsoft Technet Documentation here:

<http://technet.microsoft.com/en-us/library/jj134193.aspx>

If you do not configure the Microsoft Fax Server properly, this protocol will not work.

[Click here for steps to configure a FAXSrv carrier to use a local Microsoft Fax Server](#)

[Click her for steps to configure a FAXSrv carrier to use a network Microsoft Fax Server](#)

Local Microsoft Fax Server Configuration

PageGate can reference a Microsoft Fax Server either on its local system or, if one has been configured on the same network, another server. These steps are for configuring the local Microsoft Fax Server.

[Click here for steps on configuring a Network Microsoft Fax server.](#)

Note: Please install the [Microsoft Fax Server](#) first.

- 1) If you do not already have a Windows User Account you want PageGate to use, create one.
- 2) In Windows, grant this Account the "log on as service" policy.

PageGate Admin: How to Configure PageGate

- 3) Log out of Windows on the PageGate server, then log back in with the new user account
- 4) Open the Microsoft Fax Service Manager.
- 5) Go to Accounts.
- 6) Create a Fax Account that matches the Windows Logon Name of the PageGate account, which should also be the user account you're logged in with.
- 7) Close the Microsoft Fax Service Manager.
- 8) Run Windows Fax and Scan
- 9) Go to Tools - Fax Accounts - Add.
- 10) Select "Connect to a Fax Server on my network"
- 11) Enter \\127.0.0.1 and click on Next.

Note: When referencing a Microsoft Fax server, you MUST share the Fax in Devices and Printers, even if the Microsoft Fax Server is installed locally on the PageGate Server. If the Fax in question has not been shared, the PageGate Server will be denied access to it.

- 12) Enter a name or title for this connection, click on Done then click on Close.
- 13) From within Windows Fax and Scan, send a test fax to verify that the connection is established and functions.
- 14) If you have PageGate running as a Windows Service, apply the username and password configured for Faxing in the "Log On" section of the PageGate, PageGate Scheduler and PageGate Connector X services in the Services list. If you are running PageGate as an Application, you MUST be logged in as the Fax Account User in Windows for this to function properly.
- 15) Go through the steps in [Creating a FAXSrv Carrier](#).

The fax document is rendered by notepad.exe and the default settings are unique to the Windows User Account that PageGate is using. To modify the margins, format or output of headers and footers:

- 1) Log in to Windows as the Fax Account User.
- 2) Run Notepad in Windows.
- 3) Go to File - Page Setup.
- 4) Make any modifications you wish, then click on OK.

Network Microsoft Fax Server Configuration

Note: This can only be performed on Server based operating systems such as Server 2003, Server 2008, Server 2012 and Server 2016. Presently, according to Microsoft's specifications, you cannot share a Fax on a workstation based operating system such as Windows 7, 8 or 10.

PageGate Admin: How to Configure PageGate

6

PageGate can reference a Microsoft Fax Server either on its local system or, if one has been configured on the same network, another server. These steps are for configuring a Network Microsoft Fax server.

[Click here for steps on how to configure a Local Microsoft Fax Server.](#)

Note: The [Microsoft Fax Server](#) must be installed on the remote system for this to function. Do not install the Microsoft Fax Server locally when using this method.

On the server with the fax hardware:

1) Log in to Windows as the account named Administrator.

Note: It is not recommended to log in as any other account, even if your account is an Administrator or is part of the Domain Administrators group. The User Accounts Control in Server 2008, Server 2012 and Server 2016 requires you to log in as the account named Administrator for you to have access to certain security functions, such as configuring a Microsoft Fax Server.

2) In Windows, go to Devices and Printers.

3) Right click on Fax, left click on Printer Properties.

4) Go under the Sharing Tab.

5) Click "Change Sharing Options"

6) Put a check in "Share this printer"

7) Give the Fax a share name.

8) If you want the formatting of the faxes to be controlled by the PageGate system, leave "Render print jobs on client computers" checked.

9) Go under the Security tab.

10) Click Add.

11) Add the user you are going to use for PageGate.

Note: If you do not already have a local system or domain user for PageGate to use, please create one and grant it administrative privileges, then add that account.

There's also a bit of standardization with the user account/username that PageGate uses on its own system that will need to be done since it will need to match the name of the Fax Service account. After configuring the domain or local user account for PageGate to use, we must next identify that account as a Fax Account. To do so, please go through these steps:

1) Open the Administrative Tools in Windows.

PageGate Admin: How to Configure PageGate

- 2) Run the Fax Service Manager
- 3) On the left hand side, click on Accounts.
- 4) Make sure that a Fax Service Account that precisely and exactly matches the primary logon for the PageGate Server. This is very important because this account must match the name of the account that the PageGate services use to run, including any domain information.
- 5) If you do not presently have an Account listed that matches the PageGate account, go to Action - New - Account.
- 6) Enter the username and domain information.
- 7) Click Create to add the account in to the Fax Service Manager.

Now that we've configured the Microsoft Fax Server, we'll need to configure the PageGate server to reference it.

- 1) Log in to Windows as the Service Account that the PageGate Services are using and that the Microsoft Fax Server is referencing. It is very important to log in as that specific user in Windows on the PageGate server.
- 2) In Windows, browse over the network to the Server hosting the shared Microsoft Fax Server.
- 3) In the list of folders and items listed under the server, you should see 'Fax'. Right click on this and left click on Connect.
- 4) If prompted for credentials, provide them. This should also install the necessary fax drivers required to use this interface locally.
- 5) Open the Microsoft Fax Service Manager.
- 6) Go to Accounts.
- 7) Create a Fax Account that matches the Windows Logon Name of the PageGate account, which should also be the user account you're logged in with.
- 8) Close the Microsoft Fax Service Manager.
- 9) Run Windows Fax and Scan
- 10) Go to Tools - Fax Accounts - Add.
- 11) Select "Connect to a Fax Server on my network"
- 12) Enter the UNC path to your existing Microsoft Fax Server (\\IPorMachineName) and click on Next.

PageGate

Note: When referencing a Microsoft Fax server across the network, you MUST share the Fax in Devices and Printers on the host system. If the Fax in question has not been shared, the PageGate Server will be denied access to it.

- 13) Enter a name or title for this connection, click on Done then click on Close.
- 14) From within Windows Fax and Scan, send a test fax to verify that the connection to the remote Microsoft Fax Server is established and functions.
- 15) If you have PageGate running as a Windows Service, apply the Domain or Network Windows User Account and password configured for Faxing in the "Log On" section of the PageGate, PageGate Scheduler and PageGate Connector X services in the Services list. If you are running PageGate as an Application, you MUST be logged in as the Fax Account User in Windows for this to function properly.
- 16) Go through the steps in [Creating a FAXSrv Carrier](#).

The fax document is rendered by notepad.exe and the default settings are unique to the Windows User Account that PageGate is using. To modify the margins, format or output of headers and footers:

- 1) Log in to Windows as the Fax Account User.
- 2) Run Notepad in Windows.
- 3) Go to File - Page Setup.
- 4) Make any modifications you wish, then click on OK.

Creating a RawSerial Carrier

This protocol tells PageGate to send a Raw formatted string of data through an RS232/serial line.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the RawSerial protocol.
- 4) Enter a name in the Carrier field.
- 5) While it is unlikely that you may need to have a phone number passed as part of the RawSerial transmission, PageGate can still be configured to issue a phone number to the receiving serial device. Most implementations of this protocol will leave this field blank.
- 6) Set the Baud Rate, Parity, Data Bits and Stop Bits to match the requirements of the system or hardware that's going to be receiving this traffic.

PageGate Admin: How to Configure PageGate

7) If the receiving system requires a series of characters to be passed before it can accept input, enter that character or character string in to the Init String field.

8) Set the Max Chars field to the maximum number of characters you want output in a single transmission. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

9) Click Apply.

If the receiving system has special formatting requirements, you can configure that in the Template sub-section of a carrier.

In the Text section of the template, you can configure a string of plain text and [Macros](#) to tell PageGate what should be output and in what order. For example, if we used the following:

```
%Recipient%  
%Sender%  
%Message%
```

That tells PageGate to output the recipient's name followed by a carriage return, line feed then the recorded sender value followed by a carriage return, line feed then the body of the message in the file written.

As another example, if we used the following:

```
USER: %RecipientId%  
DATA: %Message%
```

That tells PageGate to output the text USER: before inserting the recipient's ID/PIN value followed by a carriage return, line feed then the text DATA: before inserting the body of the message to be sent.

[Click here for a list of Settings Variables that can be used with RawSerial Carriers](#)

PageGate Creating a RawTCP Carrier

This protocol tells PageGate to send a Raw formatted string of data by TCP to a specified IP address and port.

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the RawTCP protocol.
- 4) Enter a name in the Carrier field.
- 5) Enter the host IP address in the Host field.
- 6) Set the Port field to the port required by the receiving host.
- 7) If the receiving system requires a series of characters to be passed before it can accept input, enter that character or character string in to the Init String field.
- 8) Set the Max Chars field to the maximum number of characters you want output in a single transmission. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 9) Click Apply.

If the receiving system has special formatting requirements, you can configure them in the Template sub-section of a carrier.

In the Text section of the template, you can configure a string of plain text and [Macros](#) to tell PageGate what should be output and in what order. For example, if we used the following:

```
%Recipient%  
%Sender%  
%Message%
```

PageGate

That tells PageGate to output the recipient's name followed by a carriage return, line feed then the recorded sender value followed by a carriage return, line feed then the body of the message in the file written.

As another example, if we used the following:

USER: %RecipientId%

DATA: %Message%

That tells PageGate to output the text USER: before inserting the recipient's ID/PIN value followed by a carriage return, line feed then the text DATA: before inserting the body of the message to be sent.

[Click here for a list of Settings Variables that can be used with RawTCP Carriers](#)

Creating a TONE Carrier

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the TONE protocol.
- 4) Enter a name in the Carrier field.
- 5) If your modem requires a special initialization string to send DTMF, enter it in the Init String field.
- 6) Set the Max Chars field to the maximum number of digits you want to transmit in a single dial-in session.
- 7) The Wait String field determines how long the program should pause before transmitting the touch tones in the message. Your modem uses the Hayes AT command set, which means a comma represents a two second pause and an @ represents a 3 second pause.
- 8) Click Apply.
- 9) Go to Connectors - Connector X - Settings.

Note: Any connector can use a modem but a connector can only be tied to one modem at a time. Multiple connectors cannot reference the same modem.

- 10) Set the Serial Port field to match the COM port of the modem.
- 11) Click Apply.

[Click here for a list of Settings Variables that can be used with TONE Carriers](#)

PageGate Creating a UCP Carrier

- 1) Right click Carriers.
- 2) Select Add.
- 3) Select the UCP protocol.
- 4) Enter a name in the Carrier field.
- 5) Enter the phone number of the UCP terminal.
- 6) Set the Init String field to: (auto)
- 7) Set the Baud Rate, Parity, Data Bits and Stop Bits to match the receiving UCP system's requirements.
- 8) Set the Max Chars value to 160 for SMS or 240 for pagers. The maximum value of this field is 6000.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

- 9) Click Apply.
- 10) Go to Connectors - Connector X - Settings.

Note: Any connector can use a modem but a connector can only be tied to one modem at a time. Multiple connectors cannot reference the same modem.

- 11) Set the Serial Port field to match the COM port of the modem.
- 12) Click Apply.

[Click here for a list of Settings Variables that can be used with UCP Carriers](#)

Right clicking on a specific carrier will provide a context menu with four options.

- [Resend Messages](#)

This option allows you to requeue a range of previously delivered messages for the selected carrier.

- [Cancel Messages](#)

This option allows you to cancel all past due, pending and/or repeating messages for the selected carrier.

- [Pending Messages](#)

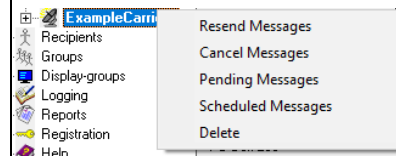
This option displays a list of all messages in the pending queue for the selected carrier.

- [Scheduled Messages](#)

This option displays a list of scheduled messages using the selected carrier.

- [Delete](#)

Selecting this option will remove the carrier from the list. This cannot be done if there are recipients still tied to this carrier.



Resend Messages

The Resend Messages option, when used at the base Carriers tree, allows you to resend any message previously sent to all carriers. If you right click on a specific carrier and select this option, it allows you to resend any message previously sent to the selected carrier.

PageGate Admin: How to Configure PageGate

6

Carrier: ExampleCarrier1

Requeue by:

- Last Message Sent
- Reliepage Number
 - Start: 1
 - End: 1
- Date & Time
 - Start: mm dd yyyy hh mm AM/PM: 3 30 2022 10 24 AM
 - End: mm dd yyyy hh mm AM/PM: 3 30 2022 10 24 AM

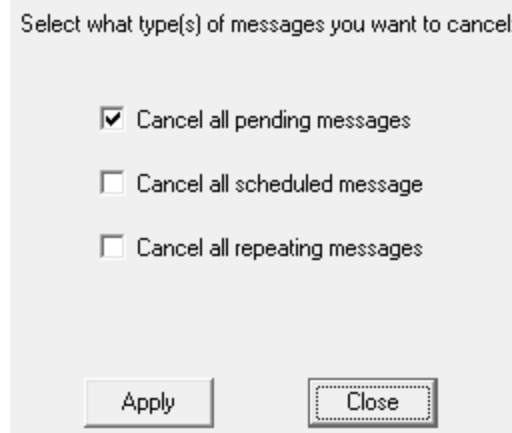
Apply

There are three options for resending messages:

- Last Message Sent
This option resends the most recent message delivered by the carrier.
- Reliepage
This option allows you to select a range of [Reliepage](#) messages to resend. The [Reliepage](#) template macro must be included in the recipient, carrier, connector, interface or global template for this to work.
- Date and Time
This option allows you to select a date and time range of messages you wish to requeue.

Cancel Messages

The Cancel Messages option, when used at the base Carriers tree, allows you to cancel messages for all carriers. If you right click on a specific carrier and select this option, it will only cancel messages for that specific carrier.



- Cancel all pending messages

Checking this option will cancel all messages in the pending queue. If a single carrier is selected, only that carrier's messages will be canceled. If the Carriers heading is selected, all messages for all carriers will be canceled.

- Cancel all scheduled messages

Enabling this option will cancel all scheduled messages. If a single carrier is selected, only that carrier's messages will be canceled. If the Carriers heading is selected, all messages for all carriers will be canceled.

- Cancel all repeating messages

Enabling this option will cancel all repeating messages scheduled by the PageGate Admin and [PageGate Client](#).

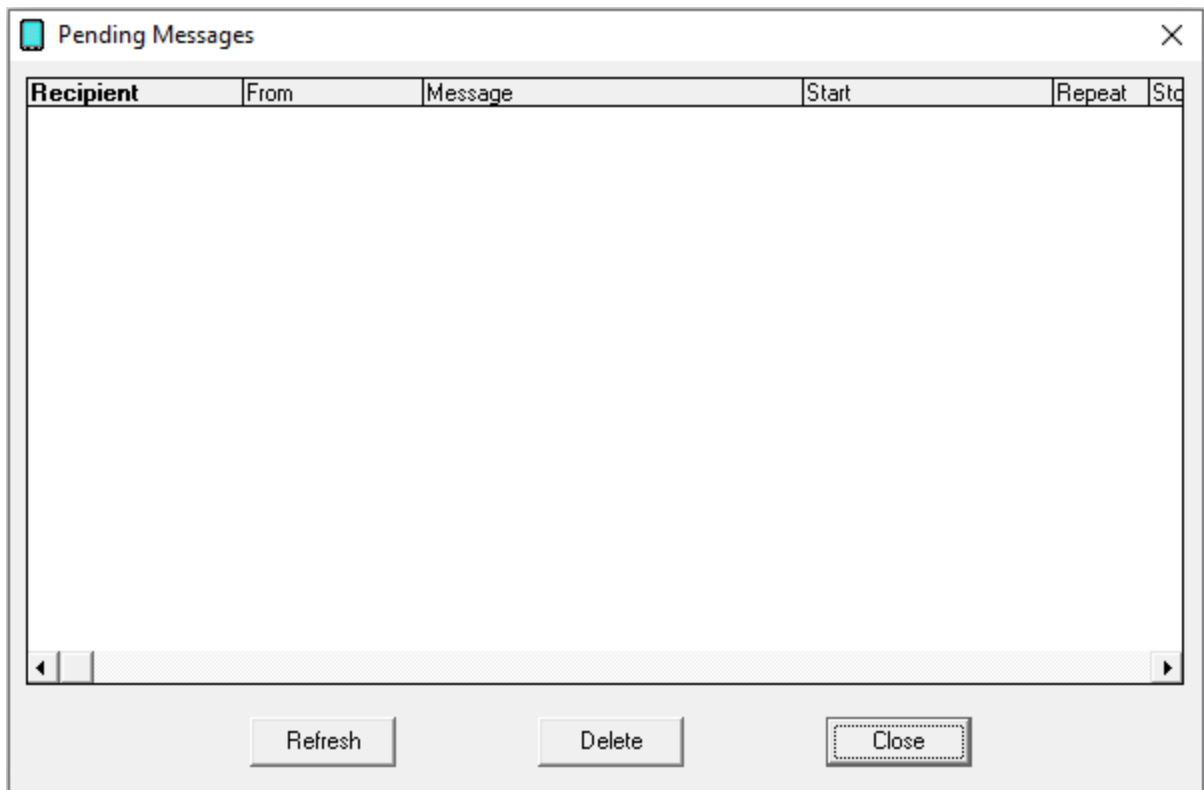
For example, if you have a message set to repeat every hour, enabling the "Cancel all repeating messages" option will stop that message from ever happening again where only enabling the "Cancel all past due message" option will only cancel any previous repeating messages that couldn't be delivered out but will not prevent future repeating messages from being sent.

PageGate Admin: How to Configure

6

PageGate Scheduled Messages

This section displays a list of all scheduled messages currently in the database.

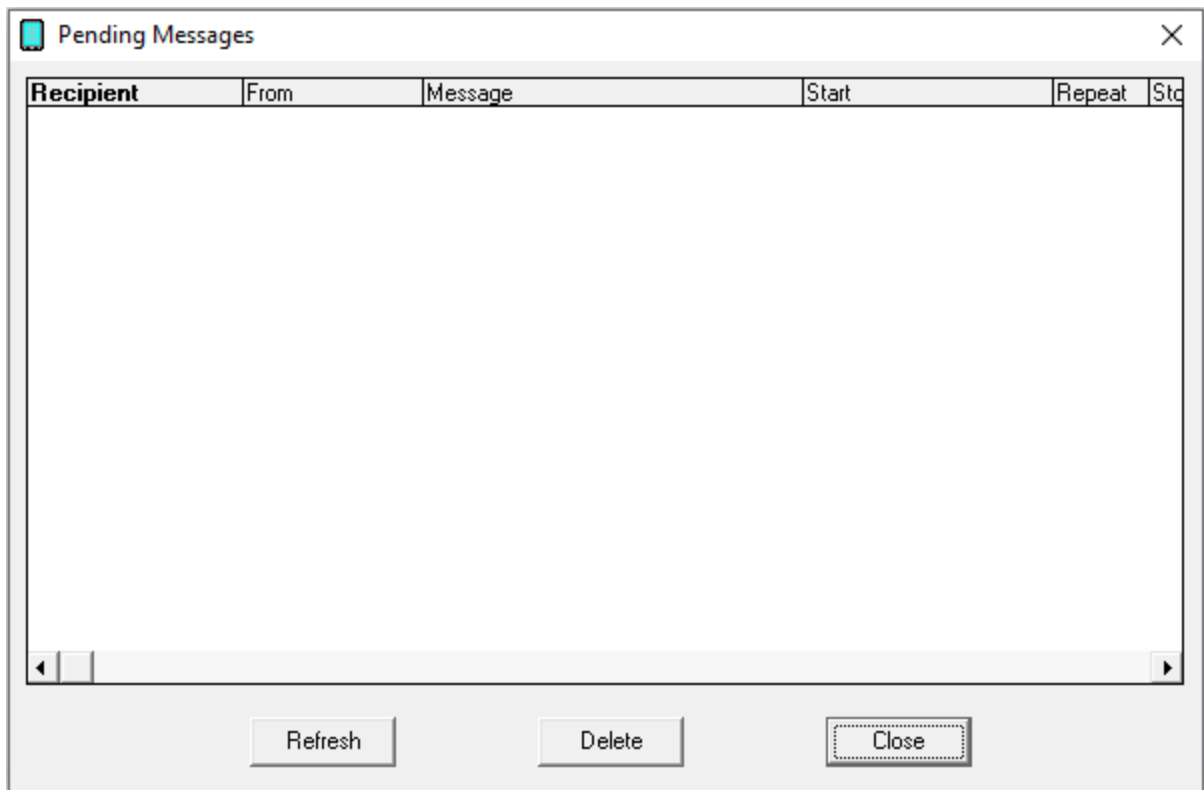


When reviewing messages, the "Pending Messages" dialog will display the following information for each Pending Message:

- Recipient
- Sender
- Content of the message
- When the schedule starts
- How often the message repeats
- When the schedule ends.

Any message displayed in this dialog may be highlighted and deleted via the Delete button.

This section displays a list of all messages currently in the pending queue.



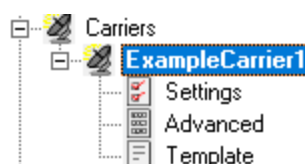
When reviewing pending messages, the "Pending Messages" dialog will display the following information for each Pending Message:

- Recipient
- Sender
- Content of the message
- When the message is to be sent
- Whether the message will repeat or only send once

Any message displayed in this dialog may be highlighted and deleted via the Delete button.

Carrier Settings

After you've gone through the steps to [create a carrier](#), they'll appear in the Carriers list. Every carrier has three sub-sections:



PageGate Admin: How to Configure

PageGate

- Settings

This section displays information about the basic configuration of this carrier, including protocol and communication parameters. This section will vary based on the protocol configured for the carrier. For more information about the carriers' basic settings, see the [Creating a new Carrier](#) section.

- [Template](#)

This section allows you to apply a custom [template](#) to this carrier.

- [Advanced](#)

This section allows you to configure failover, authentication credentials, priority level and other delivery settings. The specific options and layout of this section will vary based on the protocol the carrier is configured to use.

- [Click here](#) for AWS SNS Advanced Settings
- [Click here](#) for SMTP Advanced Settings
- [Click here](#) for SNPP Advanced Settings
- [Click here](#) for WCTP Advanced Settings
- [Click here](#) for GSM-AT Advanced Settings
- [Click here](#) for GSM-AT-IP Advanced Settings
- [Click here](#) for HTTPGET Advanced Settings
- [Click here](#) for HTTPPOST Advanced Settings
- [Click here](#) for File Advanced Settings
- [Click here](#) for XMPP Advanced Settings
- [Click here](#) for TAP Advanced Settings
- [Click here](#) for TAP-IP Advanced Settings
- [Click here](#) for FAX Advanced Settings
- [Click here](#) for FAXSrv Advanced Settings
- [Click here](#) for RawSerial Advanced Settings
- [Click here](#) for RawTCP Advanced Settings
- [Click here](#) for TONE Advanced Settings
- [Click here](#) for UCP Advanced Settings

PageGate Admin: How to Configure PageGate

Right clicking on a Carrier will provide a context menu with four options.

- [Requeue Messages](#)

Selecting this option allows you to requeue a range of previously delivered messages to the selected carrier.

- [Cancel Messages](#)

Selecting this option allows you to cancel all past due, pending and/or repeating messages for the selected carrier.

- [Pending Messages](#)

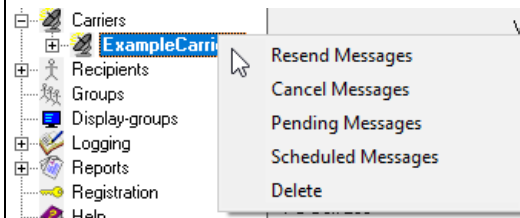
Selecting this option will display a list of all messages in the pending queue for the selected carrier.

- [Scheduled Messages](#)

Selecting this option will display a list of all messages in the pending queue for the selected carrier.

- [Delete](#)

Selecting this option will remove the carrier from the list. This cannot be done if any recipient is still tied to the carrier.



SMTP Advanced Settings

Failover	<input type="text" value="(none)"/>
Username	<input type="text"/>
Password	<input type="text"/>
Priority	<input type="text" value="100"/>
Max PIN Length	<input type="text" value="0"/>
<input checked="" type="radio"/> Deliver directly to destination server	
<input type="radio"/> Deliver through outgoing mail server	
Relay Server	<input type="text"/>
<input type="button" value="Apply"/>	

Setting	Function
---------	----------

PageGate

<p>Failover</p>	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you're delivering to WCTP, SMTP and SNPP hosts. You could configure the program to failover from WCTP to either SNPP or SMTP and then configure a second layer of failover between SMTP and SNPP. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
<p>Username</p>	<p>When using PageGate as a mail server with the 'direct delivery' option, leave this field blank.</p> <p>When relaying traffic through an SMTP server, enter a valid username as required by the SMTP server.</p> <p>For more information, see How to send Email.</p>
<p>Password</p>	<p>When using PageGate as a mail server with the 'direct delivery' option, leave this field blank.</p> <p>When relaying traffic through an SMTP server, enter a valid password as required by the SMTP server.</p> <p>For more information, see How to send Email.</p>
<p>Priority</p>	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
<p>Max PIN Length</p>	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>
<p>Deliver directly</p>	<p>Selecting this option configures PageGate to function as a mail sever. For more information, see How to send Email.</p>
<p>Deliver through</p>	<p>Selecting this options configures PageGate to relay all traffic for this carrier to the host specified in the Relay Server field. For more information, see How to send Email.</p>
<p>Apply</p>	<p>This button saves any changes made to this carrier's Advanced Settings.</p>

Failover	(none) ▾
Username	<input type="text"/>
Password	<input type="password"/>
Priority	<input type="text" value="100"/>
Msg Limit	<input type="text" value="0"/>
Max PIN Length	<input type="text" value="0"/>
<input type="button" value="Apply"/>	

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you're delivering to WCTP, SMTP and SNPP hosts. You could configure the program to failover from WCTP to either SNPP or SMTP and then configure a second layer of failover between SMTP and SNPP. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Username	If the SNPP server requires authentication, enter the username here.
Password	If the SNPP server requires authentication, enter the password here.
Msg Limit	This field determines the number of messages that can be negotiated with the receiving SNPP system in one session. Set this value to what the receiving system supports.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you're delivering to WCTP, SMTP and SNPP hosts. You could configure the program to failover from WCTP to either SNPP or SMTP and then configure a second layer of failover between SMTP and SNPP. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Username	If the WCTP host requires authentication credentials, enter the username here.
Password	If the WCTP host requires authentication credentials, enter the password here.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

Failover	<input type="text" value="(none)"/>
Username	<input type="text"/>
Password	<input type="text"/>
Priority	<input type="text" value="100"/>
Msg Limit	<input type="text" value="0"/>
Max PIN Length	<input type="text" value="0"/>
<input type="button" value="Apply"/>	

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you have two cellular modems. You could configure one cellular modem as the primary method of delivery, then configure a second carrier for the backup modem and configure the first carrier to fail over to the second. Alternately, you could configure a failover to go to a completely different method of communication.</p> <p>So, the program could initially try to negotiate SMS delivery with your cellular hardware but if the hardware is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Msg Limit	<p>This field determines the number of messages that the cellular modem can handle in one block. This setting is directly effected by the make and model of cellular modem you have as different devices have different settings and limitations. Set this field to 5 for most cellular modems but if you notice that you're running in to CMS delivery errors, lower this value as appropriate.</p>
Max PIN Length	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>

PageGate

Apply This button saves any changes made to a carrier's Advanced Settings.

GSM-AT-IP Advanced Settings

Failover	<input type="text" value="(none)"/>
Username	<input type="text"/>
Password	<input type="text"/>
Priority	<input type="text" value="100"/>
Msg Limit	<input type="text" value="0"/>
Max PIN Length	<input type="text" value="0"/>
<input type="button" value="Apply"/>	

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you have two cellular gateways. You could configure one cellular gateways as the primary method of delivery, then configure a second carrier for the backup device and configure the first carrier to fail over to the second. Alternately, you could configure a failover to go to a completely different method of communication.</p> <p>So, the program could initially try to negotiate SMS delivery with your cellular hardware but if the hardware is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Username	If your cellular hardware requires a username to send SMS, enter it here.
Password	If your cellular hardware requires a password to send SMS, enter it here.
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>

PageGate Admin: How to Configure PageGate

6

Msg Limit	This field determines the number of messages that the cellular hardware can handle in one block. This setting is directly effected by the make and model of cellular modem, router or gateway you have as different devices have different settings and limitations. It is recommended to set this value to 1 for most GSM-AT-IP carriers.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

HTTPGET Advanced Settings

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you have two separate web APIs, a primary API and a backup API. You could configure two separate HTTPGET carriers, then have one fail over to the other.</p> <p>Alternately, you could configure a failover to go to a completely different method of communication. So, the program would initially try to negotiate message delivery with your web API but if the API is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Username and Password	<p>If the Web API you're interacting with requires a login, enter the required username and password here.</p>

PageGate Admin: How to Configure PageGate

Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>
Apply	<p>This button saves any changes made to a carrier's Advanced Settings.</p>

HTTPPOST Advanced Settings

Setting	Function
<p><u>Failover</u></p>	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you have two separate web APIs, a primary API and a backup API. You could configure two separate HTTPPOST carriers, then have one fail over to the other.</p> <p>Alternately, you could configure a failover to go to a completely different method of communication. So, the program would initially try to negotiate message delivery with your web API but if the API is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>

PageGate

User name and Password	<p>Using the Username and Password in the carrier settings configures PageGate to use Basic HTTP Authentication, which is fairly standard.</p> <p>Other forms of authentication, like authorization headers or required API keys in the XML/JSON/etc body, are configured in the POST carrier's template.</p> <p>For example, another standard is to use the Authorization header which is typically in the format: Authorization: <type> <credentials></p> <p><type> and <credentials> will be defined in the receiving APIs documentation.</p>
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>
Apply	<p>This button saves any changes made to a carrier's Advanced Settings.</p>

File Advanced Settings

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host. For example, let's say that you have two separate locations that you could write files to, a primary location and a backup location. You could configure two separate File carriers, then have one fail over to the other.</p> <p>Alternatively, you could configure a failover to go to a completely different</p>

PageGate

	method of communication. So, the program would initially try to write a file to the location specified in the settings of your File carrier but if the destination is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

XMPP Advanced Settings

The screenshot shows a web form for XMPP Advanced Settings. It includes the following fields and values:

- Failover: (none) (dropdown menu)
- Username: (empty text input)
- Password: (empty text input)
- Priority: 100 (text input)
- Msg Limit: 0 (text input)
- Max PIN Length: 0 (text input)
- Apply: (button)

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, if you have multiple possible XMPP servers that you can communicate with, you can configure each as a separate carrier, then configure one to be the failover for the other. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p> <p>Alternately, you could configure a failover to go to a completely different method of communication. So, the program would initially try to negotiate a</p>

PageGate Admin: How to Configure PageGate

6

	chat session with your XMPP server but if that server is unreachable, you can have it failover to an SMTP/Email delivery method instead.
Username	Enter your XMPP server's required username.
Password	Enter your XMPP server's required password, if any.
Priority	This field determines this carrier's place within the priority system. By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.
Msg Limit	This field determines the number of messages that can be negotiated with the XMPP server in one session. Since most XMPP servers only require you to authenticate once before sending a series of messages, you can usually leave this at 0. However, if your XMPP server does require you to authenticate every message, set this field to 1.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

TAP Advanced Settings

Setting	Function
Failover	Failover is the ability of PageGate to send an undeliverable message to an alternate host. For example, some carriers support multiple TAP terminals. Other carriers support both SNPP and WCTP hosts. If you configure multiple TAP Terminals or hosts for one carrier, you can specify a failover. The failover settings at the

PageGate Admin: How to Configure PageGate

6

	carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.
Priority	This field determines this carrier's place within the priority system. By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.
Password	If the receiving TAP system requires authentication credentials, enter them in this field.
Msg Limit	This field determines the number of messages that can be negotiated with the receiving TAP system in one session. Most TAP terminals can receive up to 25 messages in a single transmission but set this value appropriate to what the receiving system supports.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Pulse Dial	If you need to use pulse dialing instead of tone dialing, check this option.
Apply	This button saves any changes made to a carrier's Advanced Settings.

TAP-IP Advanced Settings

Setting	Function
Failover	Failover is the ability of PageGate to send an undeliverable message to an alternate host. For example, some carriers support multiple TAP terminals. Other carriers support both SNPP and WCTP hosts. If you configure multiple TAP Terminals or hosts for one carrier, you can specify a failover. The failover settings at the

PageGate Admin: How to Configure PageGate

6

	carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.
Priority	This field determines this carrier's place within the priority system. By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.
Password	If the receiving TAP system requires authentication credentials, enter them in this field.
Msg Limit	This field determines the number of messages that can be negotiated with the receiving TAP system in one session. Most TAP terminals can receive up to 25 messages in a single transmission but set this value appropriate to what the receiving system supports.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Pulse Dial	If you need to use pulse dialing instead of tone dialing, check this option.
Apply	This button saves any changes made to a carrier's Advanced Settings.

FAX Advanced Settings

The screenshot shows a configuration form with the following fields:

- Failover:** A dropdown menu currently set to "(none)".
- Priority:** A text input field containing the value "100".
- Max PIN Length:** A text input field containing the value "0".
- Apply:** A button located below the input fields.

Setting	Function
Failover	Failover is the ability of PageGate to send an undeliverable message to an alternate host. For example, you could configure a Fax carrier and an SMTP/Email carrier. Then you could specify the SMTP/Email carrier as the failover for the Fax carrier and if a Fax can't be delivered, it will instead be delivered as an email. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.

PageGate

Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

FAXSrv Advanced Settings

Setting	Function
Failover	Failover is the ability of PageGate to send an undeliverable message to an alternate host. For example, you could configure a Fax carrier and an SMTP/Email carrier. Then you could specify the SMTP/Email carrier as the failover for the Fax carrier and if a Fax can't be delivered, it will instead be delivered as an email. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

Failover	<input type="text" value="(none)"/>
Username	<input type="text"/>
Password	<input type="text"/>
Priority	<input type="text" value="100"/>
Msg Limit	<input type="text" value="0"/>
Max PIN Length	<input type="text" value="0"/>
<input type="button" value="Apply"/>	

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you have two serial input lines that you could use with two separate RawSerial carriers. You could configure one as the primary method of delivery, then configure a second carrier for the backup serial line and configure the first carrier to fail over to the second.</p> <p>Alternative, you could configure a failover to go to a completely different method of communication. So, the program would initially try to negotiate a serial connection but if the serial receiver is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Username	If you need to pass a username during the RS232 negotiation, enter it here.
Password	If you need to pass a password during the RS232 negotiation, enter it here.
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>

PageGate

Msg Limit	This field determines the number of messages that the receiving serial system can handle in one block. This setting is directly effected by the receiving serial system's requirements and should be set to what the receiving system can accept.
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

RawTCP Advanced Settings

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, let's say that you have two TCP hosts that you could communicate with; one primary and one backup. You could configure each as a RawTCP carrier, then configure the first carrier to fail over to the second.</p> <p>Alternately, you could configure a failover to go to a completely different method of communication. So, the program would initially try to negotiate a TCP session with your host but if receiving host is unreachable, you can have it failover to an SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Username	If you need to pass a username during the TCP negotiation, enter it here.
Password	If you need to pass a password during the TCP negotiation, enter it here.

Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Msg Limit	<p>This field determines the number of messages than the receiving TCP host can handle in one block. This setting is directly effected by the TCP host's requirements and should be set to what the receiving system can accept.</p>
Max PIN Length	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>
Apply	<p>This button saves any changes made to a carrier's Advanced Settings.</p>

TONE Advanced Settings

The screenshot shows a web form for TONE Advanced Settings. It contains three input fields: 'Failover' is a dropdown menu set to '(none)'; 'Priority' is a text input field containing '100'; and 'Max PIN Length' is a text input field containing '0'. Below these fields is an 'Apply' button.

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, you could configure a TONE carrier and an SMTP/Email carrier. Then you could specify the SMTP/Email carrier as the failover for the TONE carrier and if a TONE message can't be delivered, it will instead be delivered as an email.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>

PageGate

Apply This button saves any changes made to a carrier's Advanced Settings.

UCP Advanced Settings

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host.</p> <p>For example, some carriers support multiple UCP terminals. Other carriers support both SNPP and WCTP hosts. If you configure multiple UCP Terminals or hosts for one carrier, you can specify a failover.</p> <p>The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p>
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Password	<p>If the receiving UCP system requires authentication credentials, enter them in this field.</p>
Msg Limit	<p>This field determines the number of messages that can be negotiated with the receiving UCP system in one session. Most UCP terminals can receive up to 25 messages in a single transmission but set this value appropriate to what the receiving system supports.</p>
Max PIN Length	<p>This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).</p>
Pulse Dial	<p>If you need to use pulse dialing instead of tone dialing, check this option.</p>
Apply	<p>This button saves any changes made to a carrier's Advanced Settings.</p>

Setting	Function
Failover	<p>Failover is the ability of PageGate to send an undeliverable message to an alternate host. The failover settings at the carrier level is used to redirect all the pending messages to another carrier when the original carrier can't be contacted.</p> <p>For example, you could configure multiple AWS SNS carriers and tie each of them to a separate host (us-east-1, us-west-2, etc) and have one fail over to the other.</p> <p>Alternately, you could configure a failover to go to a completely different method of communication. So, the program would initially try to negotiate SMS delivery with AWS SNS but if AWS is down or unreachable, you can fail over to SMTP, WCTP, SNPP, TAP or any other delivery method instead.</p>
Username	The 'Access Key ID' of an AWS account that has permission to use the AWS SNS.
Password	The 'Secret Access Key' associated with the 'Access Key ID' used in the username field.
Priority	<p>This field determines this carrier's place within the priority system.</p> <p>By default, all carriers have their priority value set to 100 which puts them all on equal footing. If you leave 100 as your baseline, giving a carrier a priority value lower than 100 will be considered more important with a value of 1 being the most important and a value higher than 100 will be considered less important up to the maximum value of 32000 being the least important.</p>
Max PIN Length	This field is used to limit the number of digits accepted from a recipient's ID/PIN field for this carrier (extra digits are stripped of the beginning).
Apply	This button saves any changes made to a carrier's Advanced Settings.

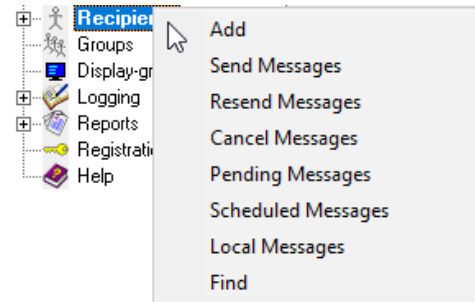
PageGate Recipients

In concept, a [carrier](#) tells PageGate how to talk to the outside world and a recipient tells PageGate who to talk to. Since the [carriers](#) tell PageGate how to talk to the outside world, they must be configured before you can create recipients.

Recipients are, generally speaking, the people, devices, hosts and/or locations you want to send messages to.

Right clicking on Recipients will provide a context menu with five options.

- [Add](#)
This option allows you to create a new recipient.
- [Send Messages](#)
This option will send a message to every recipient in the list.
- [Resend Messages](#)
This option allows you to resend a range of previously delivered messages to ALL recipients.
- [Cancel Messages](#)
This option allows you to cancel all past due, pending and/or repeating messages for ALL recipients.
- [Pending Messages](#)
This option displays a list of all messages in the pending queue for all recipients.
- [Scheduled Messages](#)
This option displays a list of all scheduled and repeating messages in the queue for all recipients.
- [Local Messages](#)
This option displays a list of local/internal messages for all recipients.
- [Find](#)
This option allows you to search for entries in the recipients list.



PageGate Admin: How to Configure PageGate

A recipient's settings can be broken down in to three sections

1) Name

Full Name	Example Recipient
Recipient	example_recipient

Full Name is the name displayed in the [GUI Client](#) and messaging websites published by the [GetWeb API](#).

Recipient is the name referenced by [the APIs](#) and is the name assigned to this recipient as a database value.

2) Enabled Services

Enabled Services		
<input type="checkbox"/> GetAscii	<input type="checkbox"/> GetMail	<input type="checkbox"/> GetSerial
<input type="checkbox"/> GetTap	<input type="checkbox"/> GetWeb	<input checked="" type="checkbox"/> GUI Client

This section displays which of [PageGate's APIs](#) are enabled for this recipient.

Services that are grayed out have not been configured and are not enabled. Services not checked are not enabled for this recipient.

For example, GetASCII isn't checked for this recipient, the GetASCII interface will not be able to message them. Similarly, if GUI Client isn't checked for this recipient, its name will not display in the GUI Client's list.

If an Interface is not available and it should be, please make sure that the Interface is enabled both in the [Program Settings](#) and in the [Interface Settings](#).

3) Specific configuration

Type	Normal
Carrier	ExampleCarrier1
Failover	(none)
ID/PIN	1
Max Chars	240
<input type="button" value="Apply"/>	

This section is where you'll tell PageGate what type of recipient this is, which delivery method it should use and what contact information to use when messaging them.

For more information, see the ['Creating a Recipient'](#) section.

Prioritizing Messages

PageGate allows you to prioritize your traffic by setting a priority value at both a carrier and recipient level and the 'Priority' field can be found in the Advanced sub-section of every carrier and recipient.

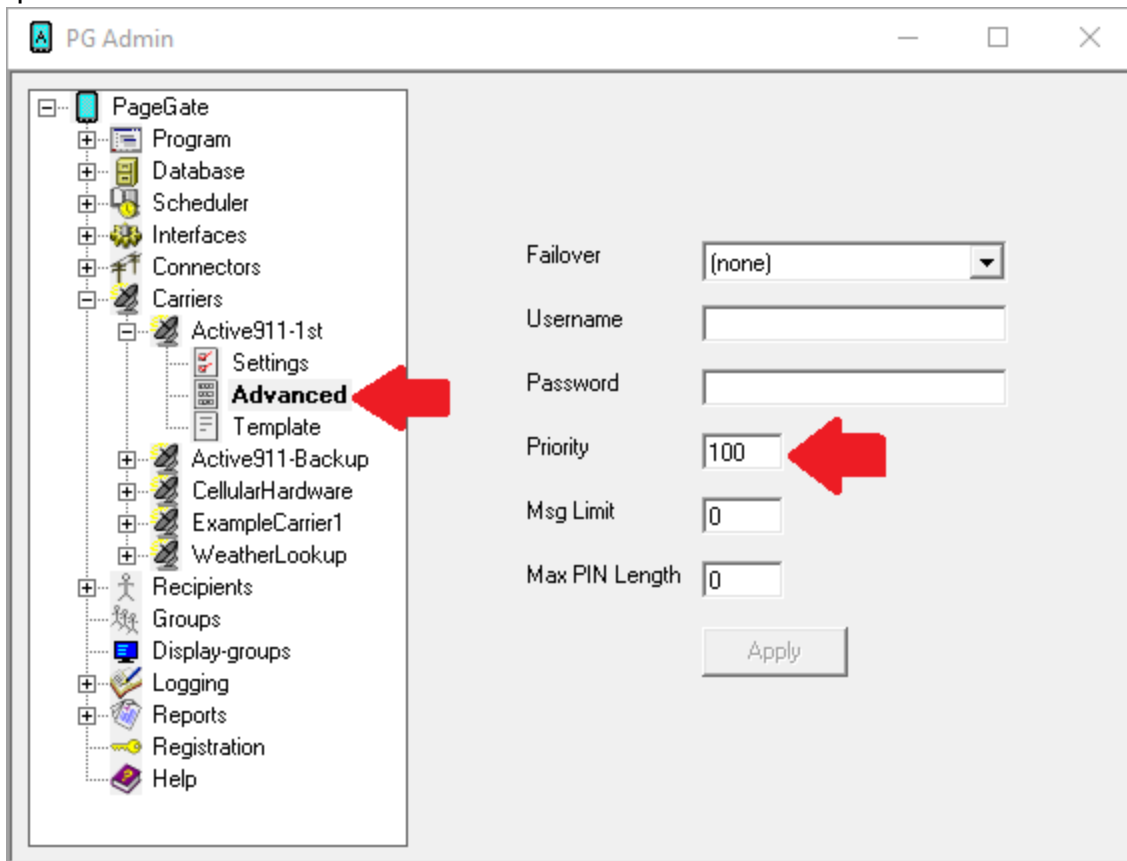
PageGate Admin: How to Configure PageGate

6

The scale of priority is 1 (the most important) to 32000 (the least important). By default, all carriers and recipients start with a priority value of 100.

Entries that share a priority value are treated equally. Entries with lower values are treated as more important, entries with higher values are treated as less important.

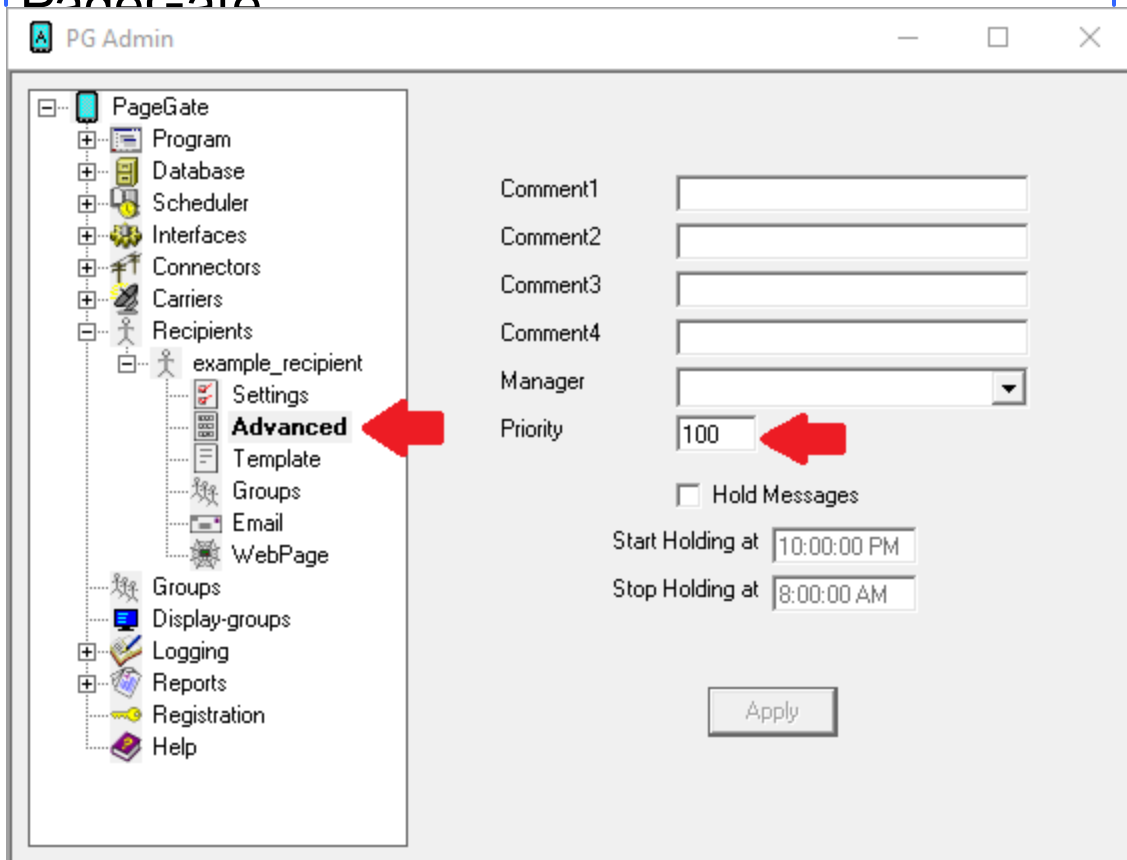
Carrier level priority determines the order in which carriers should have their messages queued.



For example, let's say that you use I Am Responding and Active911's message aggregation service and that you also deliver regular email, SMS and pages through PageGate. If you set the priority value for I Am Responding and Active911 to something lower than the priority value for the other carriers, PageGate will always schedule the I Am Responding and Active911 messages to go out before anything else.

Carrier priority values are determined in a carrier's [Advanced sub-section](#).

Recipient level priority determines the order in which recipients should receive their messages.



For example, let's say that you had a list of 100 recipients and that we had 10 recipients who needed to receive their traffic before anyone else. If you set the priority value for those recipients to something lower than the priority value for the other recipients, PageGate will always schedule their messages first. To give a slightly more practical example, let's say that you have a dispatch supervisor or fire marshal or IT director that needed to be notified before anyone else in the system. You could edit the value of that recipient's Priority field to be lower than the other recipients and that would ensure that they always received their messages first.

Recipient priority values are determined in a recipient's [Advanced sub-section](#).

Creating a Recipient

When you select the "Add" option from the [Recipient context menu](#), you'll be presented with the dialog to enter a new recipient.

PageGate Admin: How to Configure PageGate

6

Full Name

Recipient

Enabled Services

GetAscii GetMail GetSerial

GetTap GetWeb GUI Client

Type

Carrier

Failover

ID/PIN

Max Chars

Setting	Function
Full Name	This field determines the name displayed in the GUI Client and messaging websites published by the GetWeb API .
Recipient	This field determines the name assigned to this recipient as a database value. This name is also what the APIs will reference.
Enabled Services	<p>This section displays which of PageGate's APIs are enabled for this recipient.</p> <p>Services that are grayed out have not been configured and are not enabled. Services not checked are not enabled for this recipient.</p> <p>For example, if the GetASCII module isn't enabled for this recipient, that interface will not be able to message this recipient. If the GUI Client isn't enabled for this recipient, this recipient will not display in the GUI Client's list.</p> <p>If an Interface is not available and it should be, please make sure that the Interface is enabled both in the Program Settings and in the Interface Settings.</p>
Type	<ul style="list-style-type: none"> • Normal Typically speaking, you'll want to leave the type as Normal for all standard recipients; the other types are used for special configurations. • Ad-Hoc Ad-Hoc recipients function as pass-through for provided input. Put another way, an ad-hoc recipient lets you pass in the phone number or contact information you want the message delivered to rather than having that phone number exist as a static entry in the recipients list. • Email Only

	<p>If you have PageGate's GetMail API configured to host an email domain, this recipient type tells PageGate that this recipient is not a messaging recipient and cannot be messaged by the connectors. Instead, this recipient exists solely as a hosted mailbox that provides an email address that can be checked by POP.</p> <p>This type has nothing to do with the delivery of SMTP/Email and should only be used when this recipient should solely exist as a hosted mailbox.</p> <ul style="list-style-type: none">• Local This recipient type is used for internal messages to PageGate GUI Client workstations. For more information, please see Local PageGate Messaging.• Multi-Page If you have PageGate's GetMail API configured to receive email traffic, this recipient type tells PageGate that traffic sent to this recipient will contain contact information in the subject line. <p>For example, let's say you were hosting the sub-domain messaging.something.com and that you have a recipient named 'alerts' that has been configured as a Multi-Page recipient. You could then send an email to alerts@messaging.something.com with a series of 5 recipient names in the subject line and PageGate would send the body of the message to all recipients and groups referenced in the subject.</p>
Carrier	<p>This field determines the method of delivery and host being messaged. For example, if you are trying to contact a Verizon device, you would need to create a Carrier called Verizon, then set this Recipient's carrier to Verizon.</p> <p>For more information about how to configure carriers, see the How PageGate Sends Messages section.</p>
Failover	<p>If a message is not able to be delivered to this recipient, the message will be automatically redirected to whatever recipient you specify here.</p>
ID / PIN	<p>This field determines the phone number, email address or contact information for the device or host to which these messages should be delivered.</p> <p>For example, if this is an SMS recipient, you would enter the 10 digit cell phone number with no spaces or dashes or if this is an Email recipient, you would enter the email address.</p>

PageGate

Max Chars	<p>The recipient Max Chars field determines the number of characters you want to send to this person.</p> <p>If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier.</p> <p>For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 400, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.</p>
Apply	This button saves all changes made to the recipient settings.

Creating a Normal Recipient

Enter topic text here.

Creating an Ad-Hoc Recipient

Note: You must have a 5000 recipient license to unlock PageGate's ad-hoc features.

Ad-Hoc recipients function as a pass-through for provided input. Put another way, an ad-hoc recipient lets you pass in the phone number or contact information you want the message delivered to rather than having that phone number exist as a static entry in the recipients list.

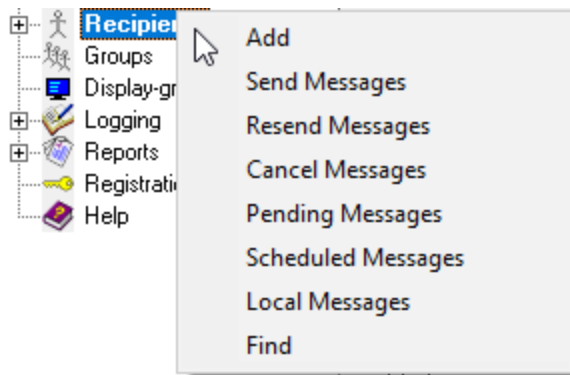
These recipients are implemented in different ways with each of PageGate's [APIs](#) and can also be a key component of creating a [two way messaging system](#).

- [Ad-Hoc for GetASCII](#)
- [Ad-Hoc for GetMail](#)
- [Ad-Hoc for GetWeb](#)
- [Ad-Hoc for GetSerial](#)
- [Ad-Hoc for GetTap](#)

Here's how to create an ad-hoc recipient:

PageGate Admin: How to Configure PageGate

1) Right click Recipients and select Add.



2) Enter a name in to the Full Name field.

Full Name

Recipient

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type

Carrier

Failover

ID/PIN

Max Chars

3) Click inside the Recipient field and make a note of the name there.

Full Name

Recipient

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type

Carrier

Failover

ID/PIN

Max Chars

Note: This is incredibly important as this specific name is what the [APIs](#) will reference.

PageGate Admin: How to Configure PageGate

6

4) Set the Type to: Ad-Hoc

The screenshot shows a configuration window for an Ad-Hoc recipient. The fields are as follows:

- Full Name: Example Ad-hoc Recipient
- Recipient: example_ad-hoc_recipient
- Enabled Services:
 - GetAscii
 - GetMail
 - GetSerial
 - GetTap
 - GetWeb
 - GUI Client
- Type: Ad-Hoc (selected in a dropdown menu)
- Carrier: ExampleCarrier1 (selected in a dropdown menu)
- Failover: (none) (selected in a dropdown menu)
- Max Chars: 1024

An "Apply" button is located at the bottom of the window.

5) In the Max Chars field, enter the maximum number of characters you want the ability to send.

Note: The Carrier Max Chars field determines the number of characters that can be received in a single message. If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier. For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.

6) Click Apply.

Ad-Hoc for GetAscii

Ad-Hoc recipients can be referenced by the [GetASCII API](#) in one of two ways.

• **Command Line reference**

When using the sendpage.exe or sendpage32.exe command line executables to pass messaging parameters to PageGate, this is how to call on an ad-hoc recipient:

```
<location of files>\sendpage32.exe ah;<value>;<value>;<value>;<value>;<etc> <sender>  
<message>
```

For example, if you have the sendpage32.exe hosted in c:\PageGateData\ASCII\ and you sent the following command:

```
c:\PageGateData\ASCII\sendpage32.exe ah;7819999998;8290500234 CAD This is a  
test message
```

PageGate

That would send a message to the phone numbers specified from the sender 'CAD' and the message would say 'This is a test message'.

- **File read**

You'll need to format the recipient section of the files being output for GetASCII to reference the ad-hoc recipient as follows:

```
ah;<value>;<value>;<value>
```

For example, let's say that you're using GetASCII's standard file format and write a test.asc file to the GetASCII polling directory formatted like this:

```
ah;<value>;<value>;<value>  
<sender>  
<message>
```

That would send the body of the message in the file to the referenced ID's from the specified sender.

Ad-Hoc for GetSerial

If you have the ability to modify the output of the system feeding data in to PageGate's [GetSerial API](#) by RS232/serial line, you can call on an ad-hoc recipient by formatting the recipient declaration contained within the serial string as follows:

```
<adhoc recipient name>;<value>;<value>;<value>
```

For example, let's say that your ad-hoc recipient's name is 'ah' and that this the serial input string GetSerial needs to process:

```
CODE 17 START ah;7818290500;9829999871 Refrigeration Unit 003 Temperature High  
84.31 N773 END A9.3178
```

Assuming that the [record](#), [recipient](#), [sender](#) and [message](#) sections have been configured appropriately, that would send a message to the two phone numbers specified.

Creating an Email Only Recipient

If you have PageGate's [GetMail API](#) configured to host an email domain, this recipient type tells PageGate that this recipient is not a messaging recipient and cannot be messaged by the connectors. Instead, this recipient exists solely as a hosted mailbox that provides an email address that can be checked by POP.

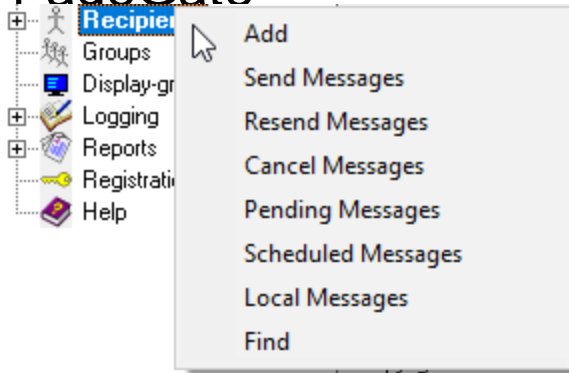
This type has nothing to do with the delivery of SMTP/Email and should **only** be used when this recipient should solely exist as a hosted mailbox.

By default, the 'Recipient' name is used as the mailbox name. However, you can change the mailbox name in the Email sub-section of this recipient.

Here's how to configure an Email Only recipient:

- 1) Right click Recipients and select Add.

PageGate Admin: How to Configure PageGate



2) Enter a name in to the Full Name field.

Full Name

Recipient

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type

Carrier

Failover

ID/PIN

Max Chars

3) Click inside the Recipient field and make a note of the name there.

Full Name

Recipient

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type

Carrier

Failover

ID/PIN

Max Chars

4) Set the Type to: Email Only.

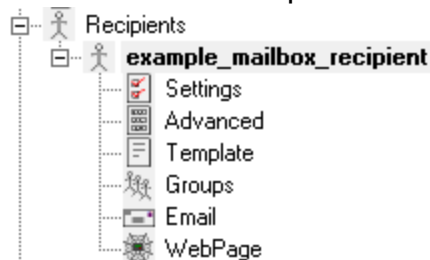
A configuration dialog box with the following fields:

- Full Name: Example Mailbox Recipient
- Recipient: example_mailbox_recipient
- Type: Email-Only (dropdown menu)
- Apply button

5) Click Apply.

To view or change the mailbox name:

1) Find the mailbox recipient in the Recipients list and click the + next to its' name.



2) Click the 'Email' sub-section of the recipient.

The PageGate Admin window is shown with the 'Recipients' folder expanded and 'example_mailbox_recipient' selected. The 'Email' sub-section is active, displaying the following configuration options:

- Mailbox Name: example_mailbox_recipient
- Forward Copy To: (empty text box)
- Enable POP Access to Messages
- POP Password: (empty text box)
- POP Mail Collection button
- Email Aliases button
- Apply button

PageGate Admin: How to Configure

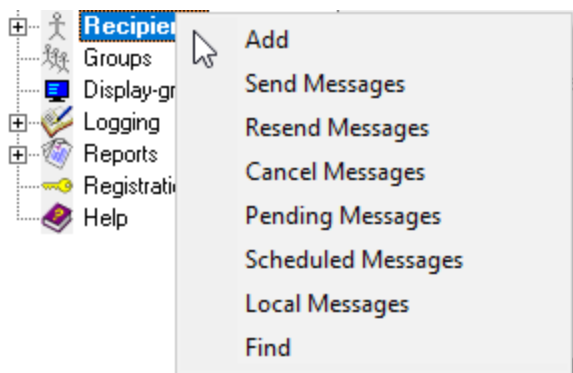
PageGate Creating a Multi-Page Recipient

If you have PageGate's [GetMail API](#) configured to receive email traffic, this recipient type tells PageGate that traffic sent to this recipient will contain contact information in the subject line.

For example, let's say you were hosting the sub-domain messaging.something.com and that you have a recipient named 'alerts' that has been configured as a Multi-Page recipient. You could then send an email to alerts@messaging.something.com with a series of 5 recipient names in the subject line and PageGate would send the body of the message to all recipients and groups referenced in the subject.

Here's how to configure a Multi-Page recipient:

- 1) Right click Recipients and select Add.



- 2) Enter a name in to the Full Name field.

Full Name	<input type="text" value="Example Multi-Page Recipient"/>
Recipient	<input type="text"/>
Enabled Services	
<input type="checkbox"/> GetAscii	<input type="checkbox"/> GetMail
<input type="checkbox"/> GetTap	<input type="checkbox"/> GetWeb
	<input checked="" type="checkbox"/> GUI Client
Type	<input type="text" value="Normal"/>
Carrier	<input type="text" value="ExampleCarrier1"/>
Failover	<input type="text" value="(none)"/>
ID/PIN	<input type="text"/>
Max Chars	<input type="text" value="1024"/>
<input type="button" value="Apply"/>	

PageGate Admin: How to Configure PageGate

3) Click inside the Recipient field and make a note of the name there.

Full Name: Example Multi-Page Recipient

Recipient: example_multi-page_recipient

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type: Normal

Carrier: ExampleCarrier1

Failover: (none)

ID/PIN:

Max Chars: 1024

Apply

Note: This is incredibly important as this specific name is what the [APIs](#) will reference.

4) Set the Type to: Multi-Page

Full Name: Example Multi-Page Recipient

Recipient: example_multi-page_recipient

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type: Multi-Page

Apply

5) Click Apply.

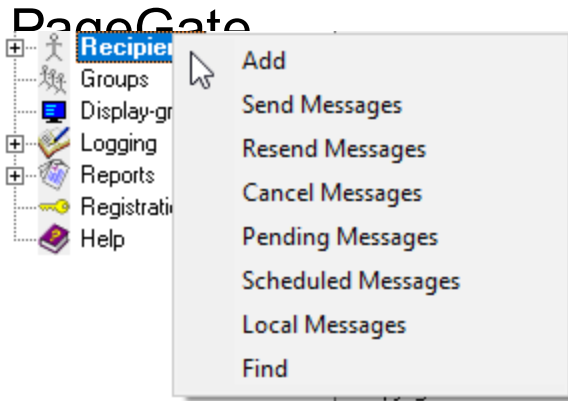
Creating a Local Recipient

Local recipients are used exclusively for messaging the PageGate GUI Client.

If a PageGate GUI Client's "Sender Name" value matches the name of a local recipient in PageGate, messages sent to that local recipient will appear in that workstation's GUI client 'Replies' window. This provides a way of sending messages between workstations and also provides the ability for GUI Client workstations to receive replies to delivered messages.

Here's how to configure a "Local" recipient for [GUI Client replies and two-way messaging](#):

1) Right click Recipients and select 'Add'.



- 2) Enter a description in the 'Full Name' field and set the 'Recipient' field to the sender's name of a GUI Client.
- 3) In the 'Type' field, select: Local

A screenshot of a configuration form. It has two text input fields: 'Full Name' containing 'Description' and 'Recipient' containing 'guisendername'. Below these is a section titled 'Enabled Services' with six checkboxes: 'GetAscii', 'GetMail', 'GetSerial', 'GetTap', 'GetWeb', and 'GUI Client'. The 'GUI Client' checkbox is checked. At the bottom is a 'Type' dropdown menu set to 'Local' and an 'Apply' button.

- 4) Click Apply.

Right clicking on a specific recipient will provide a context menu with the following options.

- [Send Messages](#)

This option will send a message to the selected recipient.

- [Resend Messages](#)

This option allows you to resend a range of previously delivered messages.

- [Cancel Messages](#)

This option allows you to cancel all past due, pending and/or repeating messages.

- [Pending Messages](#)

This option displays a list of all messages in the pending queue.

- [Scheduled Messages](#)

This option displays a list of all messages scheduled to go out at a future date and time.

- [Local Messages](#)

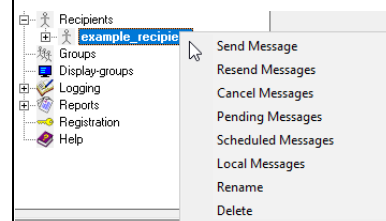
This option displays a list of all local messages.

- [Rename](#)

This option allows you to provide a new value for the selected recipient's Full Name and Recipient fields.

- [Delete](#)

This option removes the selected recipient from the list.



Send Messages

What you right click on when you select 'Send Messages' will determine who these messages go to.

For example, if you right click on the Recipients heading and select this option, this will send a message to **every** recipient in the list. If you right click on a specific recipient's name, the message will only go to the selected recipient.

PageGate Admin: How to Configure PageGate

6

The screenshot shows a web form for configuring a message. It includes the following fields and controls:

- Recipient:** A text input field containing the value "example_recipient".
- From:** An empty text input field.
- Message:** A large text area for entering the message content.
- Repeat Every:** A checkbox that is currently unchecked, followed by a text input field containing "60" and the label "minutes".
- Send:** A button located at the bottom center of the form.

Option	Description
Recipient	This field displays the recipient or group this message will go to.
From	This field determines what sender's name will be used for the message.
Message	This field determines the message that will be sent.
Repeat Every	If this message should repeat, check the box next to 'Repeat every X minutes'
Repeat Interval	If the 'Repeat Every' checkbox is checked, this determines how often the message should be sent. Fractional values of 1 minute will translate in to the appropriate number of seconds. For example, 0.5 minutes would be every 30 seconds.

Resend Messages

Selecting this option provides the ability to requeue messages delivered to all recipients or for a specific recipient. What you right click on will determine who these messages go to. For example, if you right click on the Recipients heading and select this option, this will requeue messages to every recipient in the list. If you right click on a specific recipient's name, this will requeue messages for the selected recipient.

User: example_recipient

Requeue by:

Last Message Sent

Reliepage Number

Start: 1

End: 1

Date & Time

Start: mm dd yyyy hh mm PM

End: 4 4 2022 14 42 PM

Apply

- Last Message Sent
This option resends the most recently delivered message.
- Reliepage
This option allows you to select a range of [Reliepage](#) messages to resend. The [Reliepage](#) macro must be configured in a recipient, carrier, interface, connector or global template for this to work.
- Date and Time
This option allows you to select an entire date and time range of messages you want to resend.

Cancel Messages

This option provides the ability to cancel a variety of messages. What you right click on and the options you check will determine which messages are canceled.

For example, if you right click on the Recipients heading and select this option, this will cancel messages for all recipients. If you right click on a specific recipient's name, this will only cancel messages for the selected recipient.

PageGate Admin: How to Configure PageGate

6



- Cancel all pending messages

This option cancels messages in the pending queue for immediate delivery. This will not cancel messages scheduled to go out at a future date and time.

- Cancel all scheduled messages

This option cancels messages scheduled to go out at a future date and time. This may cancel messages scheduled by a repeating message but will not delete repeating messages from going out at a later time.

For example, if you have a message set to repeat every hour, enabling the "Cancel all scheduled messages" will cancel the message for the hour you're in but the next hour's message will still queue for delivery.

- Cancel all repeating messages

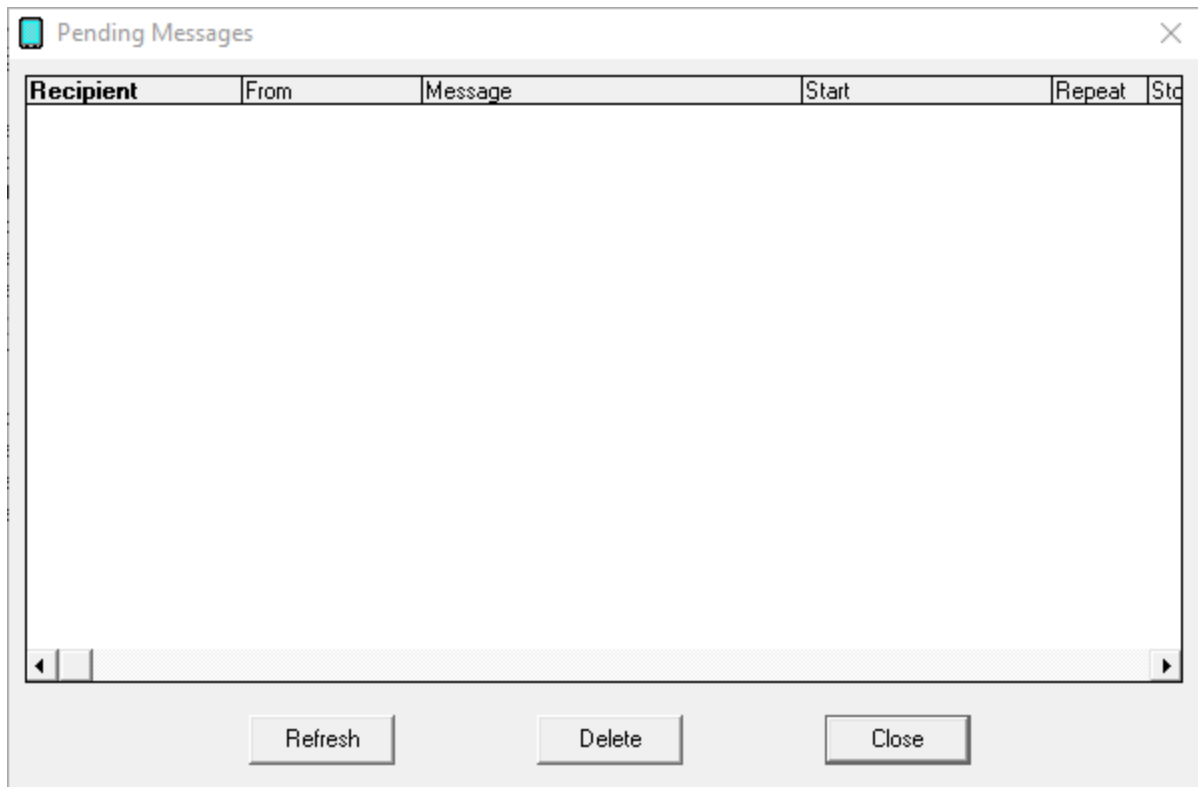
This option cancels messages scheduled to repeat.

For example, if you have a message set to repeat every hour, enabling the "Cancel all repeating messages" option will stop that message from ever happening again.

- Cancel local messages

This option cancels local messages and clears them from the GUI Client view.

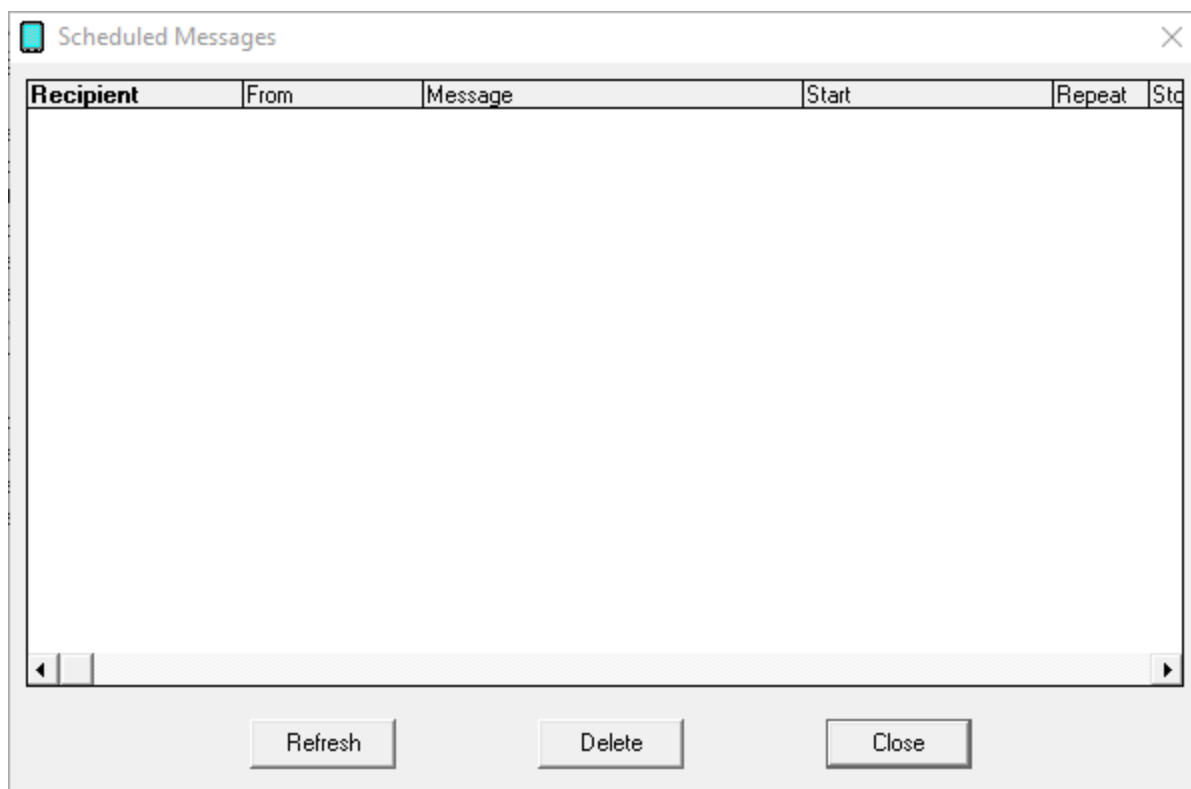
This section displays a list of all messages currently in the pending queue.



Option	Description
Recipient	This field displays the recipient or group this message will go to.
From	This field determines what sender's name was used for the message.
Message	This field determines the message to be sent.
Start	This field displays when the message entered the queue.
Repeat	If this message is scheduled to repeat, this field displays the number of minutes until the next interval.
Stop	If this is a repeating message, this field will display when the message schedule should finish. If this is not a repeating message, this field will mirror the start date and time.
Refresh	Clicking this button will refresh the list of pending messages to provide a list of current pending messages.
Delete	Selecting a message or series of messages, then clicking this button will remove them from the pending queue.
Close	This button closes the pending messages dialog.

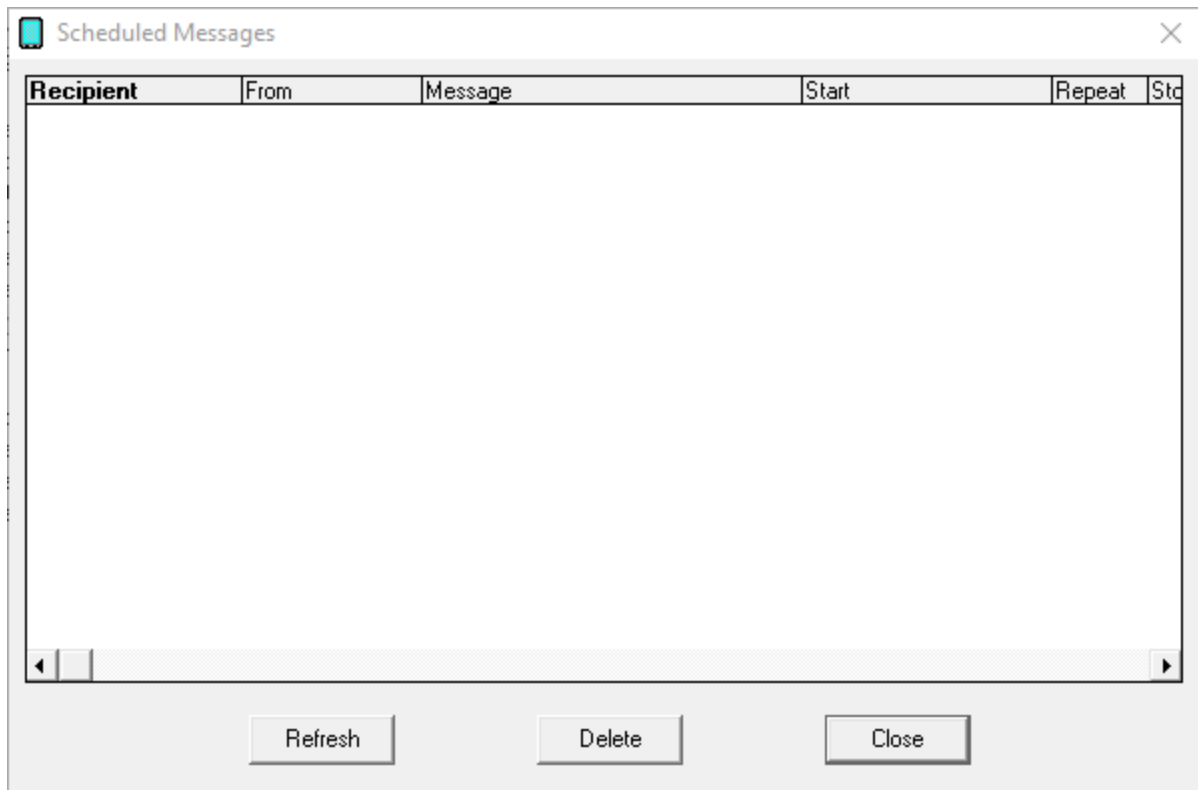
PageGate Scheduled Messages

This section displays a list of all messages scheduled to go out at a later date and time.



Option	Description
Recipient	This field displays the recipient or group this message will go to.
From	This field determines what sender's name was used for the message.
Message	This field determines the message to be sent.
Start	This field displays the date and time this message should be sent.
Repeat	If this message is scheduled to repeat, this field displays the number of minutes until the next delivery.
Stop	If this is a repeating message, this field will display when the message schedule should finish. If this is not a repeating message, this field will mirror the start date and time.
Refresh	Clicking this button will refresh the list of pending messages to provide a list of current pending messages.
Delete	Selecting a message or series of messages, then clicking this button will remove them from the pending queue.
Close	This button closes the pending messages dialog.

This section displays a list of all local messages for the selected object.



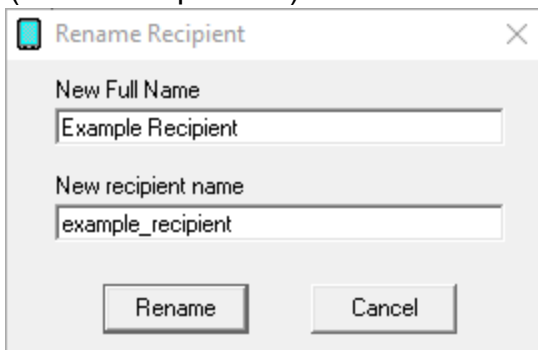
Option	Description
Recipient	This field displays the recipient or group this message will go to.
From	This field determines what sender's name was used for the message.
Message	This field determines the message to be sent.
Start	This field displays the date and time this message should be sent.
Repeat	If this message is scheduled to repeat, this field displays the number of minutes until the next delivery.
Stop	If this is a repeating message, this field will display when the message schedule should finish. If this is not a repeating message, this field will mirror the start date and time.
Refresh	Clicking this button will refresh the list of pending messages to provide a list of current pending messages.
Delete	Selecting a message or series of messages, then clicking this button will remove them from the pending queue.
Close	This button closes the pending messages dialog.

PageGate Admin: How to Configure

Rename PageGate

NOTE: Name changes do not take effect until the PageGate service is cycled. This happens automatically at midnight as part of the program's database maintenance routines but to update names immediately, you must cycle the PageGate service.

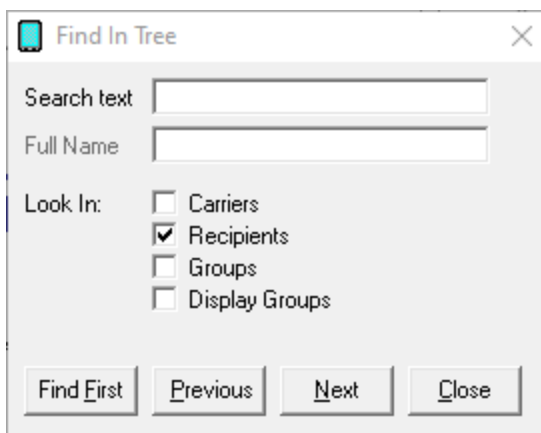
Selecting this option will provide the ability to change a recipient's display name (Full Name) or recipient name (database/api name).



The 'Rename Recipient' dialog box features a title bar with a close button. It contains two text input fields: 'New Full Name' with the value 'Example Recipient' and 'New recipient name' with the value 'example_recipient'. At the bottom are 'Rename' and 'Cancel' buttons.

Find

This option allows you to search PageGate.



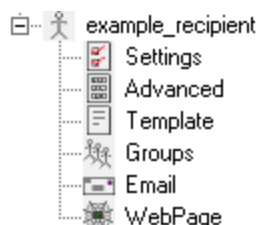
The 'Find In Tree' dialog box features a title bar with a close button. It contains two text input fields: 'Search text' and 'Full Name'. Below them are four checkboxes under 'Look In': 'Carriers' (unchecked), 'Recipients' (checked), 'Groups' (unchecked), and 'Display Groups' (unchecked). At the bottom are 'Find First', 'Previous', 'Next', and 'Close' buttons.

PageGate

Option	Description
Search text	This box searches the Recipient field (database/api name).
Full Name	This box searches the Full Name field (display name).
Look In	These checkboxes determine where the search should look.
Find First	Clicking this button will take you to the first item that matches the search text or full name.
Previous	Clicking this button will take you to the previous hit in the search.
Next	Clicking this button will take you to the next hit in the search.
Close	Clicking this button closes the Find dialog.

Recipient Settings

After you've gone through the steps to [create a recipient](#), they'll appear in the Recipients list. Clicking the + next to their name will reveal the following options:



- [Settings](#)

This section displays information about the basic configuration of this recipient, including specified carrier, contact address and enabled services. For more information, see the [Creating a Recipient](#) section.

- [Advanced](#)

This section allows you to configure [priority](#), comments and other advanced functions.

- [Template](#)

This section allows you to apply a custom template to this recipient.

- [Groups](#)

This section displays what groups, if any, this recipient is a member of and allows you to add this recipient as a member to available groups.

- [Email](#)

PageGate Admin: How to Configure

PageGate

This section is used in conjunction with the [GetMail API](#). The [GetMail API](#) must be enabled to access this section.

- [WebPage](#)

This section is used on conjunction with the [GetWeb API](#). The [GetWeb API](#) must be enabled to access this section.

Settings

Setting	Function
Full Name	This field is determines the name displayed in the GUI Client and websites published by the GetWeb API .
Recipient	This field determines the name assigned to this recipient as a database value. This name is also what the APIs will reference.
Enabled Services	<p>This section displays which of PageGate's APIs are enabled for this recipient.</p> <p>Services that are grayed out have not been configured and are not enabled. Services not checked are not enabled for this recipient.</p> <p>For example, if the GetASCII module isn't enabled for this recipient, that interface will not be able to message this recipient. If the GUI Client isn't enabled for this recipient, this recipient will not display in the GUI Client's list.</p> <p>If an Interface is not available and it should be, please make sure that the Interface is enabled both in the Program Settings and in the Interface Settings.</p>

Type	<ul style="list-style-type: none"> • Normal • Ad-Hoc • Email Only • Local • Multi-Page <p>Typically speaking, you'll want to set the type to Normal for all standard recipients; the other types are used for special configurations.</p>
Carrier	<p>This field determines the method of delivery for this recipient's messages and can only be set to a carrier configured within the PageGate Admin.</p> <p>For example, if you're trying to contact a Verizon device, you would need to create a Carrier named Verizon, then set this recipient's carrier value to Verizon.</p> <p>For more information about how to configure carriers, see the How PageGate Sends Messages section.</p>
Failover	<p>If a message is not able to be delivered to this recipient, the message will be automatically redirected to whatever recipient you specify here.</p>
ID / PIN	<p>This field determines the phone number, email address or contact information for the device or host to which these messages should be delivered.</p>
Max Chars	<p>The recipient Max Chars field determines the number of characters you want to send to this person.</p> <p>If the recipient Max Chars value is set higher than the carrier Max Chars value, PageGate will break up long messages based on the character limitation in the carrier.</p> <p>For example, if you have the carrier configured for 160 Max Chars and you set the recipient Max Chars value to 3000, then send a 400 character message to this recipient, PageGate will break up the 400 character count message in to two 160 character messages and one 80 character message. When the device receives the message, it will be prefixed with part 1/3, 2/3, 3/3 to indicate the multi-part nature of the message delivered.</p>
Apply	<p>Apply saves all recently made changes to a recipient's settings.</p>

Setting	Function
Comment1 - Comment4	These four fields are stored in each recipient. If the GUI Client Interface is used, the information in these fields can be viewed by the GUI Client operators and their displayed recipient lists can be filtered by this information.
Manager	If GUI Client Security has been enabled, this is the user or usergroup that has administrative control over this recipient.
Priority	Priority determines the order in which messages are handled. This value may be set from 1 to 32000, where 1 is the highest priority and 32000 is the lowest priority.
Hold Messages	Enabling this option will configure PageGate to hold messages for this recipient during the time specified in the Start Holding and Stop Holding entries.
Start Holding at	If Hold Messages is enabled, set this to when the program should start holding messages in queue for this recipient.
Stop Holding at	If Hold Messages is enabled, set this to when the program should send all held messages to this recipient.
Apply	This button saves all recent changes made to a recipient's Advanced settings.

Groups

The groups sub-section of the Recipient allows you to specify which groups this recipient is a member of.

The screenshot shows a configuration window with two main sections: "Non-member" and "Member". The "Non-member" section contains a list with one entry, "example_group". The "Member" section is currently empty. At the bottom of the window, there is a "Copy from" dropdown menu and an "Apply" button.

To add this recipient to the member list of a group, click on the name of the group under the "Non-member" category.

To remove this recipient from the member list of a group, click on the name of the group under the "Member" category.

Email

NOTE: To access the Email sub-section of a Recipient, the [GetMail API](#) must be configured and enabled.

This section controls mailbox hosting and email forwarding for this recipient.

The screenshot shows the "Email" configuration section. It includes a "Mailbox Name" field with the value "example_group", a "Forward Copy To" field, a checkbox for "Enable POP Access to Messages" which is unchecked, and a "POP Password" field. Below these fields are three buttons: "POP Mail Collection", "Email Aliases", and "Apply".

PageGate Admin: How to Configure PageGate

6

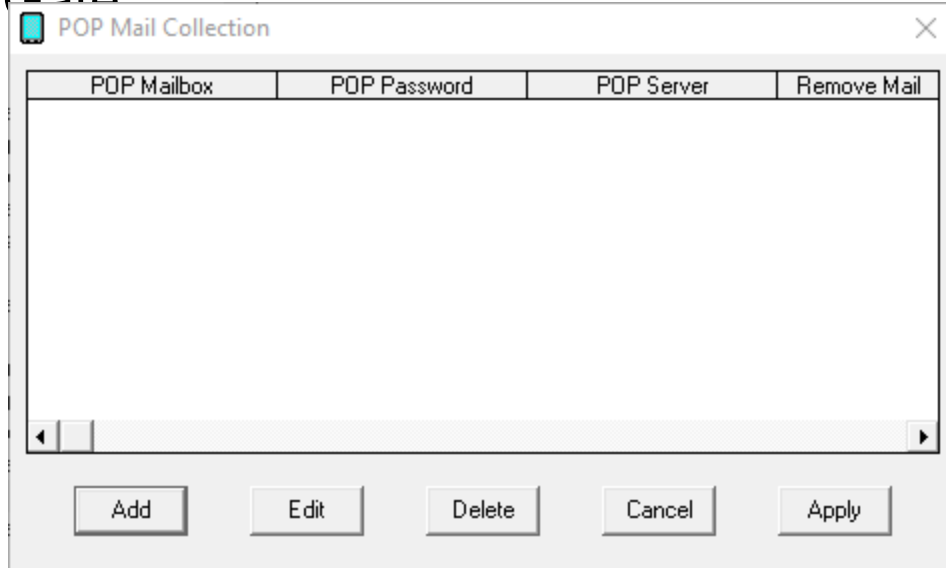
Setting	Function
Mailbox Name	<p>This field determines the hosted mailbox name of this recipient. This is the value that is used when sending an email in to PageGate for processing.</p> <p>For example, let's say that we have the GetMail API hosting a sub-domain of messaging.something.com. The mailbox name would be the part before the @ of any email that you wanted to send in. For instance, you would use example_recipient@messaging.something.com with the example recipient shown above.</p>
Forward Copy To	If an email address is entered in to this field, any message queued for this recipient will also be forwarded to the specified email address.
Enable POP Access to Messages	This option enables POP access to this recipient's hosted mailbox. This allows email client programs access to email hosted by the PageGate server for this mailbox.
POP Password	This option is only available when "Enable POP Access to Messages" is enabled. This field sets the password for the hosted mailbox for this recipient.
POP Mail Collection	This section allows you to tie this recipient to an externally hosted POP mailbox.
Email Aliases	This option is only available when "Enable POP Access to Messages" is enabled. Click the Email Aliases button to add, edit, or remove an Email Alias.
Apply	This button saves any recent changes to the Email section of the Recipient.

POP Mail Collection

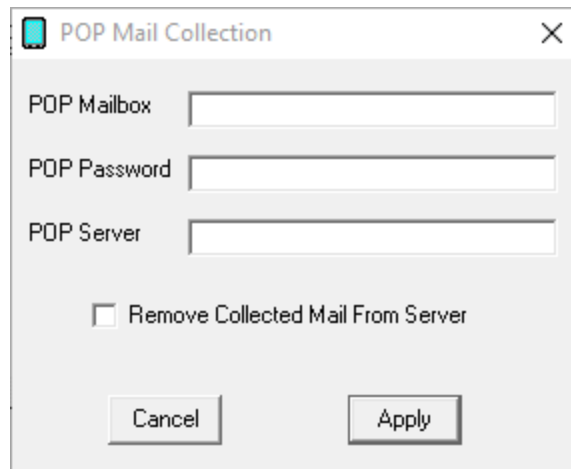
This function configures PageGate to check an externally hosted, POP accessible mailbox. Any mail found in the box will be read in and delivered to the recipient this POP Mail Collection rule exists for.

PageGate Admin: How to Configure PageGate

6



Option	Function
Add	Adds a new POP Mail Collection rule. See below.
Edit	Edits an existing POP Mail Collection rule
Delete	Removes an existing POP Mail Collection rule.
Cancel	Cancel all recent changes.
Apply	Applies new changes and rules.



PageGate

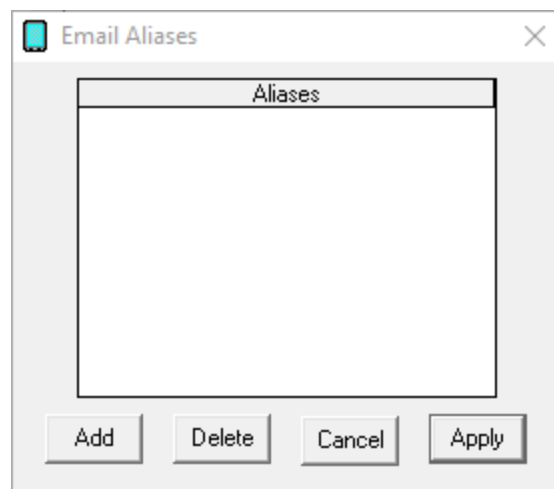
Setting	Function
POP Mailbox	Enter the mailbox name as required by the POP server. Some POP servers require the fully qualified email address, others will only require the mailbox name while others may require a domain declaration prefixed to the mailbox name. Please follow the specification of your POP server.
POP Password	Enter the password required to check the email address.
POP Server	Enter the POP3 server address for the POP3 server hosting the mailbox.
Remove Collected Mail From Server	Enabling this option will remove collected mail from the mailbox. If this option is not enabled, the mail will be marked as read but it will not be removed from the mailbox.
Cancel	Discards all recent changes to this POP Mail Collection rule.
Apply	Saves all recent changes to this POP Mail Collection rule.

Email Aliases

Each PageGate recipient can accept messages from more than one local email address and these additional names are called 'aliases'.

For example, let's say we're working with a recipient named 'station14'. Technically, that mailbox can only receive traffic addressed to `station14@yourdomain.com`. However, aliases allow you to tell PageGate that it should accept other names as if they were this mailbox.

If you want PageGate to send all messages addressed to WestPalmAlerts@yourdomain.com to station14, you would create an alias for WestPalmAlerts for the station14 recipient.



PageGate Admin: How to Configure PageGate

6

Button	Function
Add	This adds a new entry in to the Aliases table.
Delete	Highlight an existing alias and click on Delete to remove that alias.
Cancel	Clicking this button will remove all recent changes to the Email aliases.
Apply	Clicking this button will save all recent changes to the Email aliases.

Web Page

NOTE: To access the WebPage sub-section of a Recipient, the [GetWeb API](#) must be configured and enabled.

This section allows you to build a messaging website based on this recipient. Typically speaking, this is only done with [Ad-Hoc recipients](#) that need to be tied to the [GetWeb API's Ad-Hoc webpaging](#) function.

Setting	Function
Template File	<p>The GetWeb API comes with a range of templates that can be used 'out of the box' or you can modify them any way you please as long as you retain the appropriate CGI submission variables and statements. This option allows you to select which web messaging template should be applied to this recipient.</p> <p>These files associated with the templates are located in the 'templates' directory beneath the Web Pages Path directory configured in the GetWeb API.</p>

PageGate Admin: How to Configure PageGate

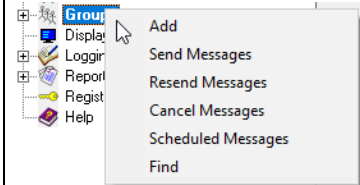
6

Text1 - Text10	<p>When implementing a messaging website, you can use <code>~~~Text1~~~</code> through <code>~~~Text10~~~</code> as variables in the HTML code of the template tied to this recipient. These fields represent the static values that will be inserted in to these variables as PageGate comes across them when building the content for the web server to host.</p> <p>For example, if you have a statement like this in your code: Welcome to the messaging website for <code>~~~Text1~~~</code>.</p> <p>and you use 'District 1' in the Text1 field, PageGate will create a webpage that says: Welcome to the messaging website for District 1.</p>
Image1 - Image5	<p>Note: Image files should be placed in the 'images' directory beneath the Web Pages Path directory configured in the GetWeb API.</p> <p>When implementing a messaging website, you can use <code>~~~Image1~~~</code> through <code>~~~Image5~~~</code> as variables in the HTML code of the template tied to this recipient. These fields represent the static values that will be inserted in to these variables as PageGate comes across them when building the content for the web server to host.</p> <p>For example, if you have a statement like this in your code: <code></code></p> <p>and you use 'customimage.jpg' in the Image1 field, PageGate will create a webpage that references the specified graphic.</p>
Custom Page	Enable this option if you aren't using one of our templates or have modified one of our templates and have saved it under a different name.
Apply	This button saves any recent changes to the WebPage section of the Recipient.

Groups are collections of [recipients](#) in the PageGate Admin.

Right clicking on Recipients will provide a context menu with four options.

- [Add](#)
This option creates a new group.
- [Send Messages](#)
This option sends a message to every group in the list.
- [Resend Messages](#)
This option resends a range of previously delivered messages to all groups.
- [Cancel Messages](#)
This allows you to cancel pending, scheduled and/or repeating messages for all groups.
- [Scheduled Messages](#)
This option displays a list of all scheduled and repeating messages for all groups.
- [Find](#)
This option allows you to search for entries in the recipients list.



Creating a Group

Description	<input type="text"/>	
Group	<input type="text"/>	
Enabled Services		
<input type="checkbox"/> GetAscii	<input checked="" type="checkbox"/> GetMail	<input type="checkbox"/> GetSerial
<input type="checkbox"/> GetTap	<input checked="" type="checkbox"/> GetWeb	<input checked="" type="checkbox"/> GUI Client
Advanced Options		
<input type="checkbox"/> On-Call Group		
<input type="text" value="On-Call Schedule"/>		
<input type="button" value="Apply"/>		

PageGate Admin: How to Configure PageGate

6

Setting	Function
Description	This is the name displayed in the GUI Client and messaging websites published by the GetWeb API .
Group	This is the group's database name and is the name the APIs reference.
Enabled Services	<p>This section displays which of PageGate's APIs are enabled for this group.</p> <p>Services that are grayed out have not been configured and are not enabled. Services not checked are not enabled for this group.</p> <p>For example, if the GetASCII module isn't enabled for this group, that interface module will not be able to message this group. If the GUI Client module isn't enabled for this group, it will not display in the GUI Client's list.</p> <p>If an Interface is not available and it should be, please make sure that the Interface is enabled both in the Program Settings and in the Interface Settings.</p>
On-Call Group	Enabling this options allows you to configure an on-call schedule for members of this group. This option should be enabled if the members of this group should only be paged during certain times and/or on certain days of the week. Please refer to the On-Call Settings for more information.
On-Call Schedule	When On-Call group is checked, this button allows you to configure the on-call schedule. Please refer to the On-Call Settings for more information.
Apply	Apply saves all recently made changes to a Group's settings.

Right clicking on a specific group will provide a context menu with five options.

- [Send Messages](#)

This sends a message to the selected group.

- [Resend Messages](#)

This resends a range of previously delivered messages to the selected group.

- [Cancel Messages](#)

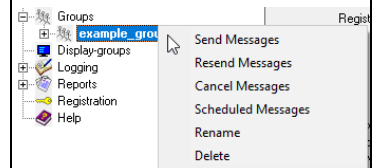
This option allows you to cancel past due, pending and/or repeating messages for the selected group.

- [Rename](#)

This allows you to rename a group's description or database name.

- [Delete](#)

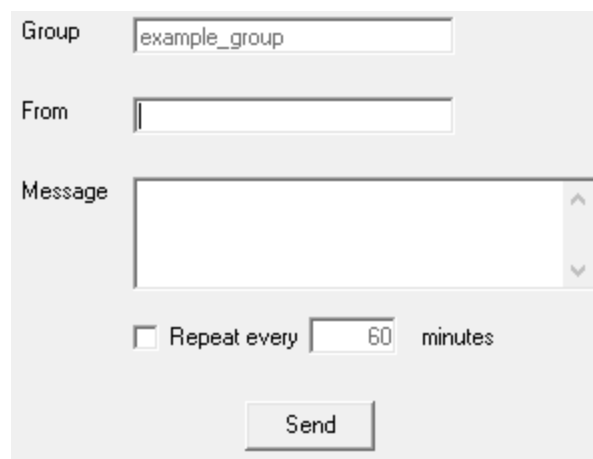
This option removes the selected group from the program.



Send Messages

What you right click on when you select 'Send Messages' will determine who these messages go to.

For example, if you right click on the Groups heading and select this option, this will send a message to **every** group in the list. If you right click on a specific group's name, the message will only go to the selected group.



PageGate

Option	Description
Recipient	This field displays who this message will go to.
From	This field determines what sender's name will be used.
Message	This field determines the message that will be sent.
Repeat Every	If this message should repeat, check the box next to 'Repeat every X minutes'
Repeat Interval	If the 'Repeat Every' checkbox is checked, this determines how often the message should be sent. Fractional values of 1 minute will translate in to the appropriate number of seconds. For example, 0.5 minutes would be every 30 seconds.

Resend Messages

What you right click on will determine who these messages go to. Selecting this option provides the ability to requeue messages delivered to a group or all groups.

For example, if you right click on the Groups heading and select this option, this will resend messages to every group in the list. If you right click on a specific group's name, this will resend messages for the selected group.

Group:

Requeue by:

Last Message Sent

Reliepage Number

Start:

End:

Date & Time

Start:

End:

- Last Message Sent

This option resends the most recently delivered message.

PageGate

- Date and Time

This option allows you to select an entire date and time range of messages you want to resend.

Note: Reliepage requeuing is not available for groups.

Cancel Messages

This option provides the ability to cancel a variety of messages. What you right click on and the options you check will determine which messages are canceled.

For example, if you right click on the Groups heading and select this option, this will cancel messages for all groups. If you right click on a specific group's name, this will only cancel messages for the selected group.



- Cancel all pending messages

This option cancels messages in the pending queue for immediate delivery. This will not cancel messages scheduled to go out at a future date and time.

- Cancel all scheduled messages

This option cancels messages scheduled to go out at a future date and time. This may cancel messages scheduled by a repeating message but will not delete repeating messages from going out at a later time.

For example, if you have a message set to repeat every hour, enabling the "Cancel all scheduled messages" will cancel the message for the hour you're in but the next hour's message will still queue for delivery.

- Cancel all repeating messages

This option cancels messages scheduled to repeat.

For example, if you have a message set to repeat every hour, enabling the "Cancel all repeating messages" option will stop that message from ever happening again.

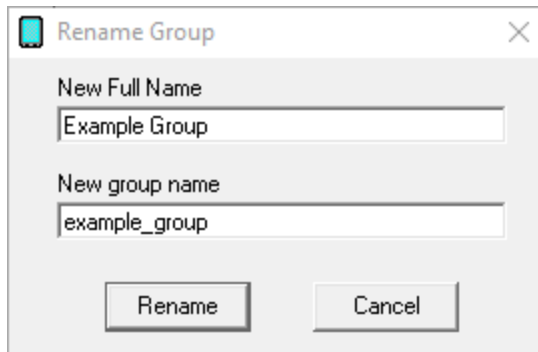
PageGate Admin: How to Configure

6

Rename PageGate

NOTE: Name changes do not take effect until the PageGate service is cycled. This happens automatically at midnight as part of the program's database maintenance routines but to update names immediately, you must cycle the PageGate service.

This provides the ability to change a group's display name (Full Name) or group name (database/api name).



Rename Group

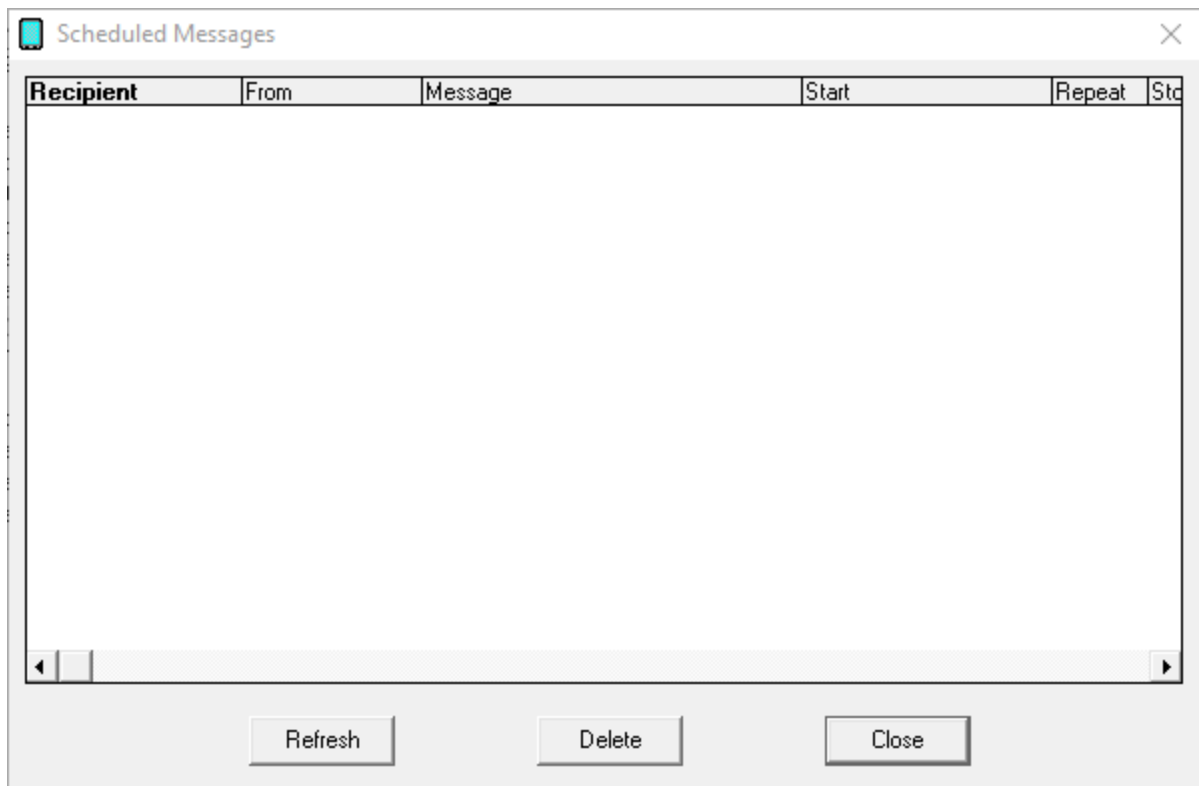
New Full Name
Example Group

New group name
example_group

Rename Cancel

Scheduled Messages

This section displays a list of all messages scheduled to go out at a later date and time.



Scheduled Messages

Recipient	From	Message	Start	Repeat	Std
-----------	------	---------	-------	--------	-----

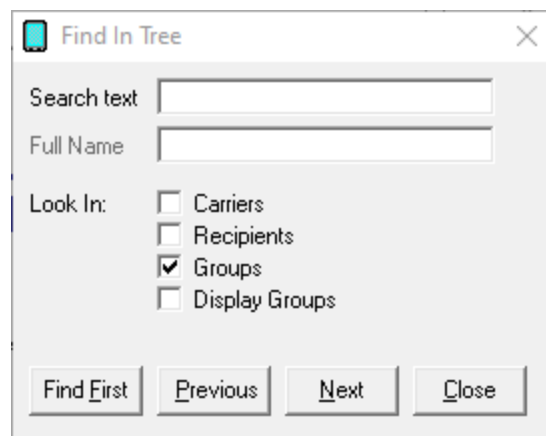
Refresh Delete Close

PageGate

Option	Description
Recipient	This field displays the recipient or group this message will go to.
From	This field determines what sender's name was used for the message.
Message	This field determines the message to be sent.
Start	This field displays the date and time this message should be sent.
Repeat	If this message is scheduled to repeat, this field displays the number of minutes until the next delivery.
Stop	If this is a repeating message, this field will display when the message schedule should finish. If this is not a repeating message, this field will mirror the start date and time.
Refresh	Clicking this button will refresh the list of pending messages to provide a list of current pending messages.
Delete	Selecting a message or series of messages, then clicking this button will remove them from the pending queue.
Close	This button closes the pending messages dialog.

Find

This option allows you to search PageGate.



PageGate

Option	Description
Search text	This box searches the group field (database/api name).
Full Name	This box searches the Full Name field (display name).
Look In	These checkboxes determine where the search should look.
Find First	Clicking this button will take you to the first item that matches the search text or full name.
Previous	Clicking this button will take you to the previous hit in the search.
Next	Clicking this button will take you to the next hit in the search.
Close	Clicking this button closes the Find dialog.

Group Settings

After you've gone through the steps to [create a group](#), they'll appear in the Groups list. Every recipient has five sub-sections:



- [Settings](#)

This section displays information about the basic configuration of this recipient, including specified carrier, contact address and enabled services. For more information, see the [Creating a Recipient](#) section.

- [Advanced](#)

This section allows you to configure comments, security manager and message holds.

- [Members](#)

This section allows you to apply a custom template to this recipient.

- [Email](#)

This section is used in conjunction with the [GetMail API](#). The [GetMail API](#) must be enabled to access this section.

- [WebPage](#)

PageGate Admin: How to Configure

PageGate

This section is used in conjunction with the [GetWeb API](#). The [GetWeb API](#) must be enabled to access this section.

Settings

Description: Example Group
Group: example_group

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

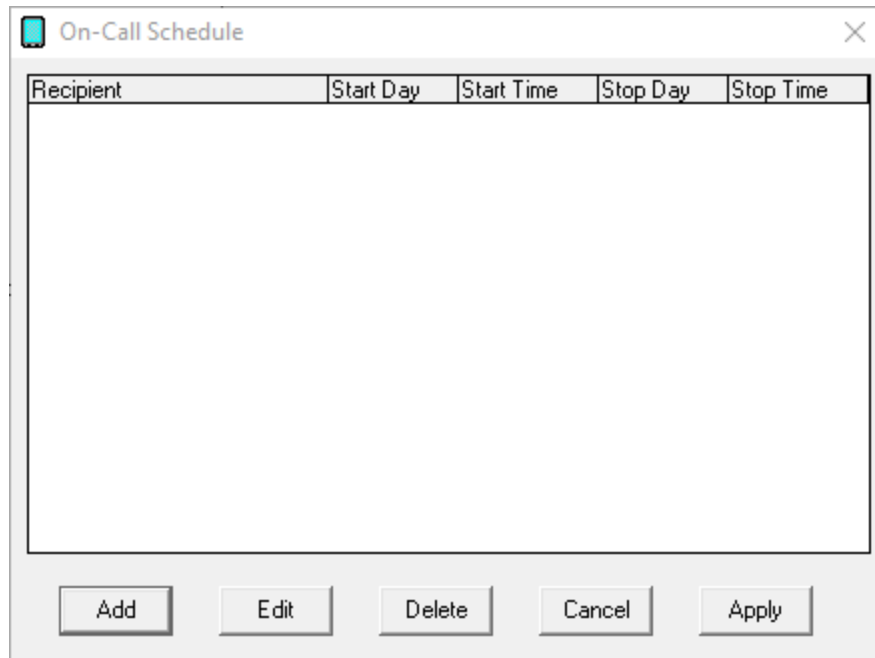
Advanced Options

On-Call Group

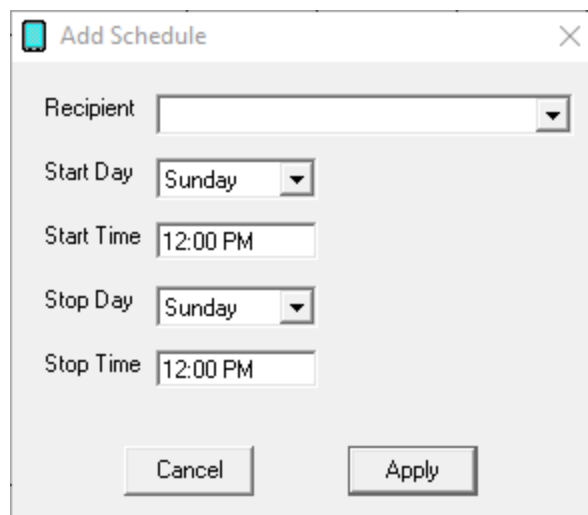
On-Call Schedule

Apply

Setting	Function
Description	This field determines the name displayed in the GUI Client and messaging websites published by the GetWeb API .
Group	This field determines the name assigned to this group as a database value. This name is also specifically what the APIs will reference.
Enabled Services	<p>This section displays which of PageGate's APIs are enabled for this group.</p> <p>Services that are grayed out have not been configured and are not enabled. Services not checked are not enabled for this group.</p> <p>For example, if the GetASCII module isn't enabled for this group, that interface module will not be able to message this group. If the GUI Client module isn't enabled for this group, it will not display in the GUI Client's list.</p> <p>If an Interface is not available and it should be, please make sure that the Interface is enabled both in the Program Settings and in the Interface Settings.</p>
On-Call Group	Enabling this options allows you to configure an on-call schedule for members of this group. This option should be enabled if the members of this group should only be paged during certain times and/or on certain days of the week. Please refer to the On-Call Settings for more information.
On-Call Schedule	When On-Call group is checked, this button allows you to configure the on-call schedule. Please refer to the On-Call Settings for more information.
Apply	Apply saves all recently made changes to a Group's settings.



Option	Function
Add	Adds a new rule to the on-call schedule. See below.
Edit	Edits an on-call schedule rule
Delete	Removes an existing on-call schedule rule.
Cancel	Cancel all recent changes.
Apply	Applies new changes and rules.



Option	Function
Recipient	This field specifies which recipient this on-call schedule rule applies to.

PageGate

Start Day	This field determines what day of the week this on-call rule begins.
Start Time	This field determines what time of the day this on-call rule begins.
Stop Day	This field determines what day of week this on-call rule ends.
Stop Time	This field determines what time of the day this on-call rule ends.

Advanced

Setting	Function
Comment1 - Comment4	These four fields are stored in each group. If the GUI Client Interface is used, the information in these fields can be viewed by the GUI Client operators and their displayed group lists can be filtered by this information.
Manager	If GUI Client Security has been enabled, this is the user that has administrative control over this group.
Hold Messages	Enabling this option will configure PageGate to hold messages for this group during the time specified in the Start Holding and Stop Holding entries.
Start Holding at	If Hold Messages is enabled, set this to when the program should start holding messages in queue for this group.
Stop Holding at	If Hold Messages is enabled, set this to when the program should send all held messages to this group.
Apply	This button saves all recent changes made to a group's Advanced settings.

Members

The Members sub-section of a Group allows you to specify which recipients and groups are a member of the selected group.

PageGate Admin: How to Configure PageGate

6

The screenshot shows a web interface with two main panels: 'Non-members' on the left and 'Members' on the right. The 'Non-members' panel contains a list with one entry, 'example_recipient'. The 'Members' panel is currently empty. Below these panels is a 'Copy from' dropdown menu and an 'Apply' button.

To add an entry to the Members list, click on the name of the recipient or group under the "Non-members" category.

To remove an entry from Members the list, click on the name of the recipient or group under the "Members" category.

To copy another group's membership, select that group in the 'Copy from' field.

Email

NOTE: To access the Email sub-section of a Group, [GetMail](#) **must** be configured and enabled.

This section controls mailbox hosting and email forwarding for this group.

PageGate Admin: How to Configure PageGate

6

Mailbox Name: example_group

Forward Copy To: []

Enable POP Access to Messages

POP Password: []

POP Mail Collection

Email Aliases

Apply

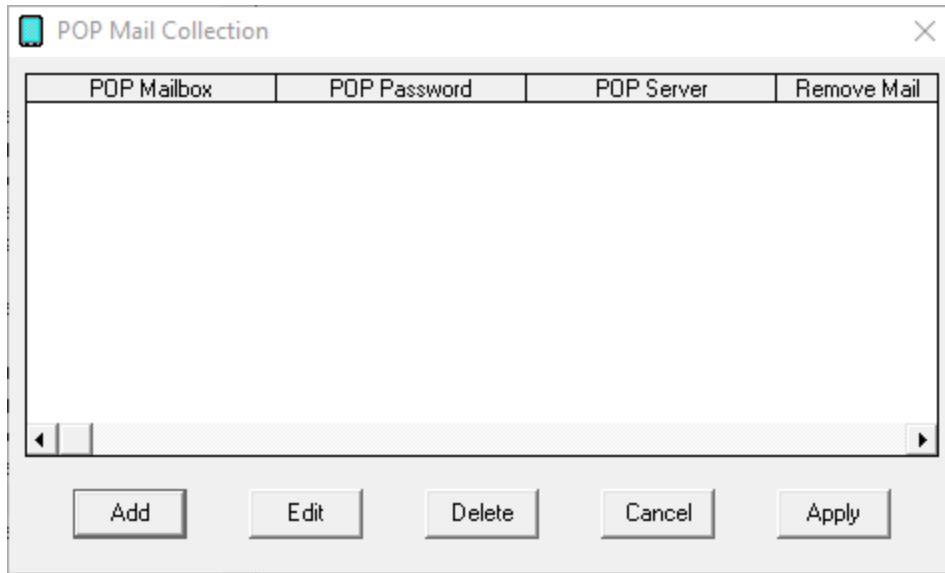
Setting	Function
Mailbox Name	This field determines the hosted mailbox name of this group. This is the value that is used when sending an email in to PageGate for processing. For example, let's say that we have the GetMail API hosting a sub-domain of messaging.something.com. The mailbox name would be the part before the @ of any email that you wanted to send in. For instance, you would use example_group@messaging.something.com with the example group shown above.
Forward Copy To	If an email address is entered in to this field, any message queued for this group will also be forwarded to the specified email address.
Enable POP Access to Messages	This option enables POP access to this group's hosted mailbox. This allows email client programs access to email hosted by the PageGate server for this mailbox.
POP Password	This option is only available when "Enable POP Access to Messages" is enabled. This field sets the password for the hosted mailbox for this group.
POP Mail Collection	This section allows you to tie this group to an externally hosted POP mailbox.
Email Aliases	This option is only available when "Enable POP Access to Messages" is enabled. Click the Email Aliases button to add, edit, or remove an Email Alias.
Apply	This button saves any recent changes to the Email section of the Recipient.

POP Mail Collection

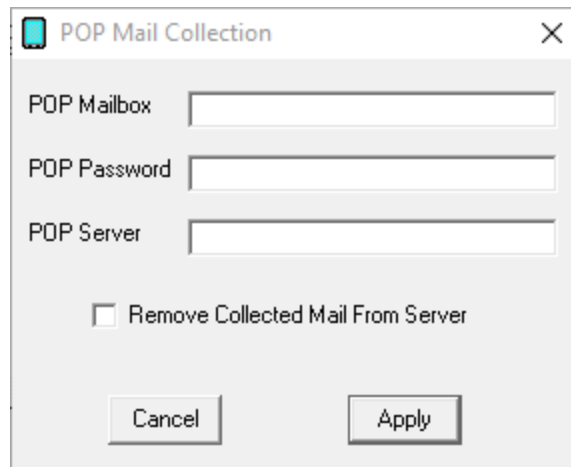
This function configures PageGate to check an externally hosted, POP accessible mailbox. Any mail found in the box will be read in and delivered to the members of the group this POP Mail Collection rule exists for.

PageGate Admin: How to Configure PageGate

6



Option	Function
Add	Adds a new POP Mail Collection rule. See below.
Edit	Edits an existing POP Mail Collection rule
Delete	Removes an existing POP Mail Collection rule.
Cancel	Cancel all recent changes.
Apply	Applies new changes and rules.



PageGate

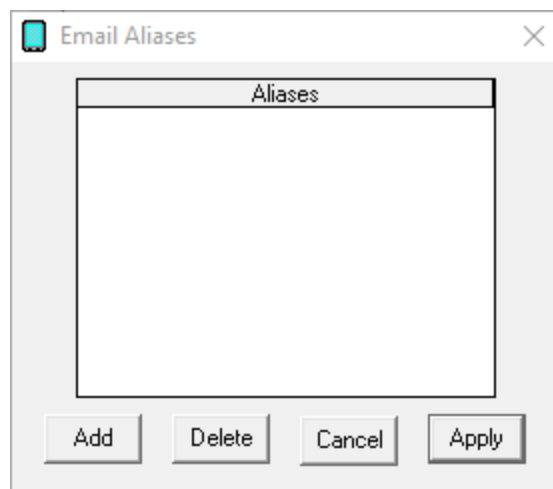
Setting	Function
POP Mailbox	Enter the mailbox name as required by the POP server. Some POP servers require the fully qualified email address, others will only require the mailbox name while others may require a domain declaration prefixed to the mailbox name. Please follow the specification of your POP server.
POP Password	Enter the password required to check the email address.
POP Server	Enter the POP3 server address for the POP3 server hosting the mailbox.
Remove Collected Mail From Server	Enabling this option will remove collected mail from the mailbox. If this option is not enabled, the mail will be marked as read but it will not be removed from the mailbox.
Cancel	Discards all recent changes to this POP Mail Collection rule.
Apply	Saves all recent changes to this POP Mail Collection rule.

Email Aliases

Each group can accept messages from more than one local email address and these additional names are called 'aliases'.

For instance, let's say we're working with a group named 'st14'. Technically, that mailbox can only receive traffic addressed to `st14@yourgetmaildomain.com`. However, aliases allow you to tell PageGate that it should accept other names as if they were this mailbox.

As an example, if you wanted PageGate to send messages addressed to WestPalmAlerts@yourdomain.com to your st14 group, you would create an alias for WestPalmAlerts for the st14 group.



Button	Function
Add	
Delete	
Cancel	
Apply	

PageGate Admin: How to Configure PageGate

6

Add	This adds a new entry in to the Aliases table.
Delete	Highlight an existing alias and click on Delete to remove that alias.
Cancel	Clicking this button will remove all recent changes to the Email aliases.
Apply	Clicking this button will save all recent changes to the Email aliases.

WebPage

NOTE: [GetWeb](#) must be configured and enabled to access this section.

This section allows you to build a messaging website based on this group's member list. To do so, you'll need to through the [GetWeb's GUI List Template](#) configuration.

Setting	Function
Template File	<p>The GetWeb API comes with a range of templates that can be used 'out of the box' or you can modify them any way you please as long as you retain the appropriate CGI submission variables and statements. This option allows you to select which web messaging template should be applied to this group.</p> <p>These files associated with the templates are located in the 'templates' directory beneath the Web Pages Path directory configured in the GetWeb API.</p>

Text1 - Text10	<p>When implementing a messaging website, you can use <code>~~~Text1~~~</code> through <code>~~~Text10~~~</code> as variables in the HTML code of the template tied to this group. These fields represent the static values that will be inserted in to these variables as PageGate comes across them when building the content for the web server to host.</p> <p>For example, if you have a statement like this in your code: Welcome to the messaging website for <code>~~~Text1~~~</code>.</p> <p>and you use 'District 1' in the Text1 field, PageGate will create a webpage that says: Welcome to the messaging website for District 1.</p>
Image1 - Image5	<p>Note: Image files should be placed in the 'images' directory beneath the Web Pages Path directory configured in the GetWeb API.</p> <p>When implementing a messaging website, you can use <code>~~~Image1~~~</code> through <code>~~~Image5~~~</code> as variables in the HTML code of the template tied to this group. These fields represent the static values that will be inserted in to these variables as PageGate comes across them when building the content for the web server to host.</p> <p>For example, if you have a statement like this in your code: <code></code></p> <p>and you use 'customimage.jpg' in the Image1 field, PageGate will create a webpage that references the specified graphic.</p>
Custom Page	Enable this option if you aren't using one of our templates or have modified one of our templates and have saved it under a different name.
Group List	This option should be checked for any group tied to the list.htm and guilist.htm templates or any derivative thereof.
Full Names	Enable this option if you want any referenced recipients' full names to appear on the webpage.
Apply	This button saves any recent changes to the WebPage section of the group.

Display Groups

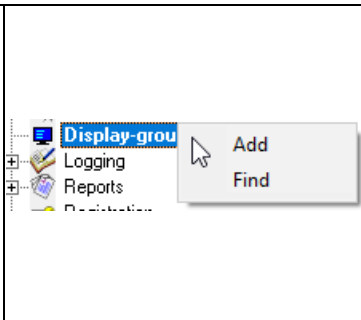
The Display Groups function is only used in conjunction with the [GUI Client API](#).

Display Groups have nothing to do with normal messaging groups. Instead, these groups allow administrators to control which recipients and groups the workstations are allowed to see.

For example, let's say that you have 20 workstations that have the GUI client installed. Let's also say that 10 of them are for dispatching fire calls and that 10 of them are for dispatching police calls. In this instance, you would want to create two display groups, one

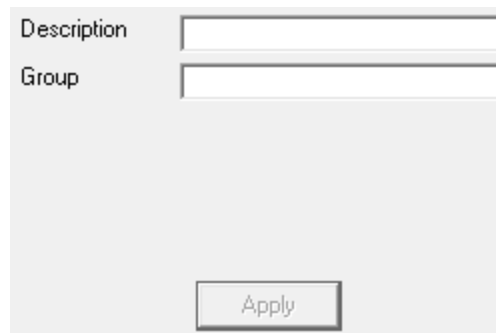
PageGate Admin: How to Configure PageGate

named 'fire' and the other named 'police' and populate them with the appropriate recipients. Once that's done, you can then tie each workstation to an appropriate display group, that way they can't message recipients and groups they aren't allowed to see.

<p>Right clicking on Display Groups will provide a context menu with the following options:.</p> <ul style="list-style-type: none"> • Add This option allows you to create a new display group. • Find This option allows you to search the display groups list. 	
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

After creating a display group, you'll need to populate it with [members](#). Any recipient or group that is a member of the display group will display on any GUI Clients tied to that display group. Any recipient or group that is not a member will not display.

Creating a Display Group



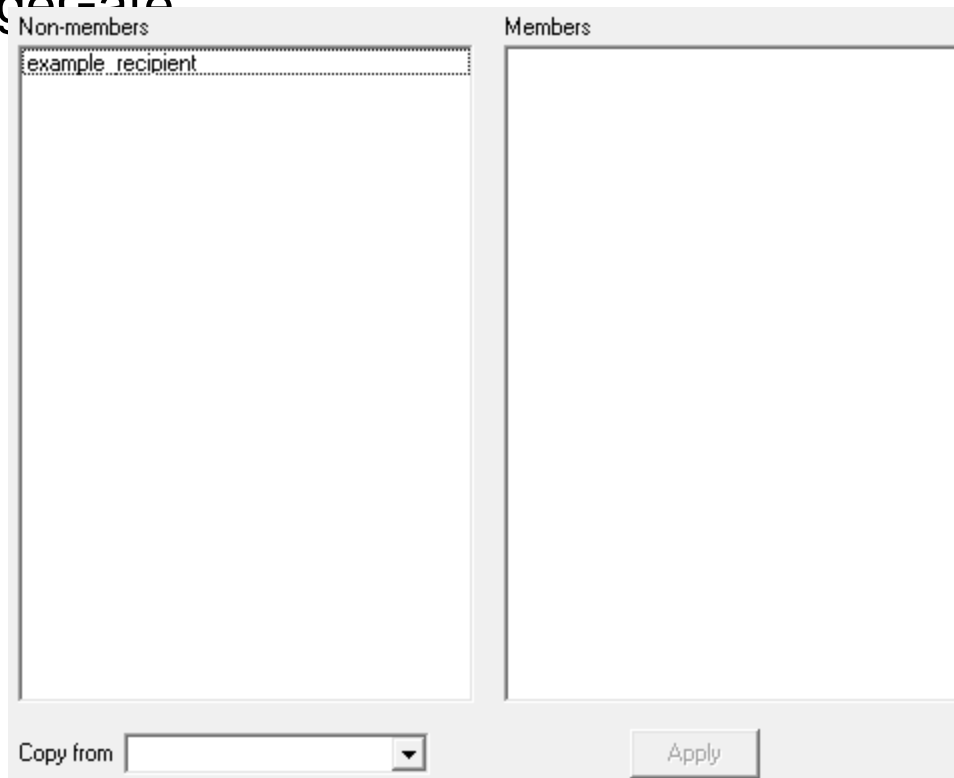
Setting	Function
Description	This field should describe the display group.
Group	This is the name used when selecting a display group in the GUI Client.
Apply	This adds the display group in to the list.

Members

The Members sub-section of a display group allows you to specify which recipients and groups are members of the selected display group.

PageGate Admin: How to Configure PageGate

6



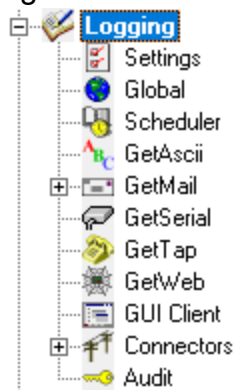
To add an entry to the Members list, click on the name of the recipient or group under the "Non-members" category.

To remove an entry from Members the list, click on the name of the recipient or group under the "Members" category.

To copy another group's membership, select that group in the 'Copy from' field.

Logging

This section contains information logged by PageGate and controls how much information PageGate should log, whether the program should archive its logs and other settings.



- [Settings](#)

PageGate Admin: How to Configure

PageGate

This section determines whether PageGate should log information, whether it should archive the logs, what type of information to keep in the logs. This section also administrates the [PG Monitor](#).

- [Global](#)

This section displays a summary of the events generated by PageGate, including all messages scheduled and sent.

- [Scheduler](#)

This section displays a summary of the events generated by the PageGate Scheduler.

- [GetASCII](#)

This section displays a record of information and traffic processed by the GetASCII API.

- [GetMail](#)

This section displays a record of information and traffic processed by the GetMail API.

- [GetSerial](#)

This section displays a record of information and traffic processed by the GetSerial API.

- [GetTap](#)

This section displays a record of information and traffic processed by the GetTap API.

- [GetWeb](#)

This section displays a record of information and traffic processed by the GetWeb API.

- [GUI Client](#)

This section displays a record of information and traffic processed by the GUI Client API.

- [Connectors](#)

This section displays a record of all information and traffic processed by the Connectors.

- [Audit](#)

This is a record of every administrative function performed by any PageGate Admin.

PageGate Admin: How to Configure PageGate

Right-clicking on Logging will provide a context menu with two option.

- Open Folder

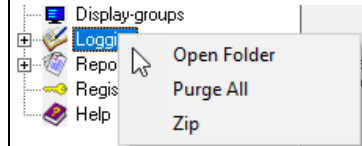
Selecting this option will the location of the log folder in Windows Explorer.

- Purge All

Selecting this option will delete all current log files.

- Zip

This option creates a zip file of your PageGate database and places it in the [database directory](#).



Settings

Log to Disk	<input checked="" type="checkbox"/>
Logging Level	Debug
Archive Logs	<input checked="" type="checkbox"/>
Max Log Length	1000000 (characters)
Use PG Monitor	<input checked="" type="checkbox"/>
PG Monitor Port	10101
PG Monitor Host0	127.0.0.1
PG Monitor Host1	
PG Monitor Host2	
PG Monitor Host3	
Apply	

Setting	Function
Log to Disk	When this option is checked, all events are written to the log files. Unchecking this option disables PageGate's logging.
Logging Level	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this tells PageGate to only keep track of the date, time, sender, recipient and message information in the logs. The program will not record protocol negotiation sequences. • <u>Detail</u> Detail will write the specific commands sent and received but will not include a full breakdown of all characters sent and received in the protocol negotiation sequences.

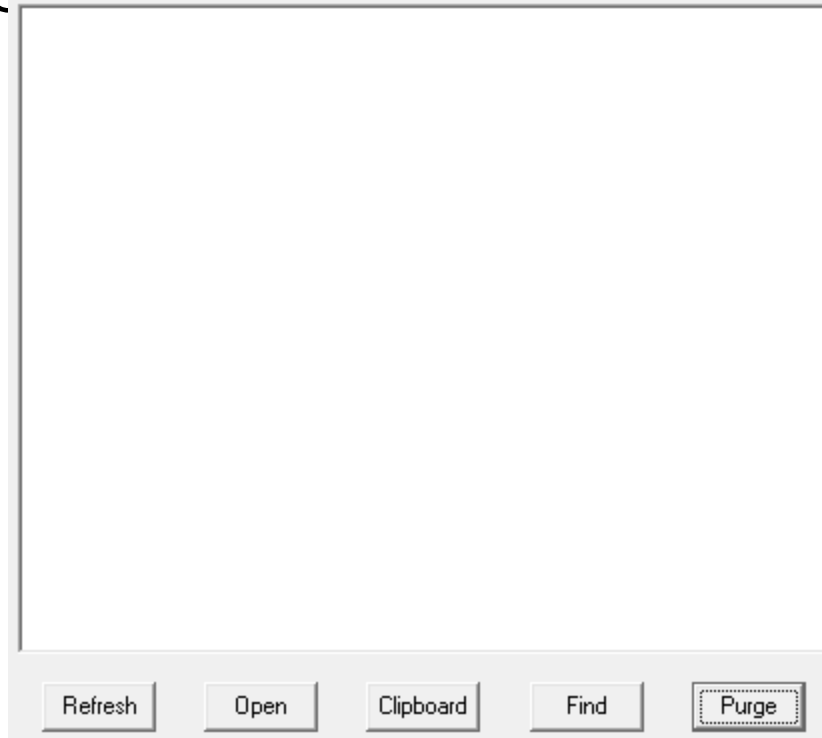
PageGate Admin: How to Configure PageGate

6

	<ul style="list-style-type: none">• <u>Debug (Recommended)</u> Selecting this will keep track of every command and response issued from and received to PageGate.
Archive Logs	Enabling this option will allow each day's logs to be archived. If this is enabled, every day at midnight, PageGate will move the previous day's logs to a sub-directory of the Logging Directory and name it the date of the archive.
Max Log Length	This field specifies how large the individual log files can become before old information is purged. If 'Archive Logs' is checked, this setting is ignored.
Use PG Monitor	This option specifies whether information is sent to the PageGate Monitor or not. If this option is not enabled, the PageGate Monitor will not display any information.
PG Monitor Port	This field specifies the TCP port the PageGate server broadcasts its monitoring information on. This field should match the PG Monitor port setting in the PageGate Monitor .
PG Monitor Host1 - Host3	These fields allow you to specify what additional systems can use the PageGate Monitor. Enter the LAN IP address or machine name of the systems you want PageGate to broadcast monitoring information to.
Apply	This saves all changes made.

Global

This log shows a summary of the events generated by PageGate, including all messages scheduled and sent.



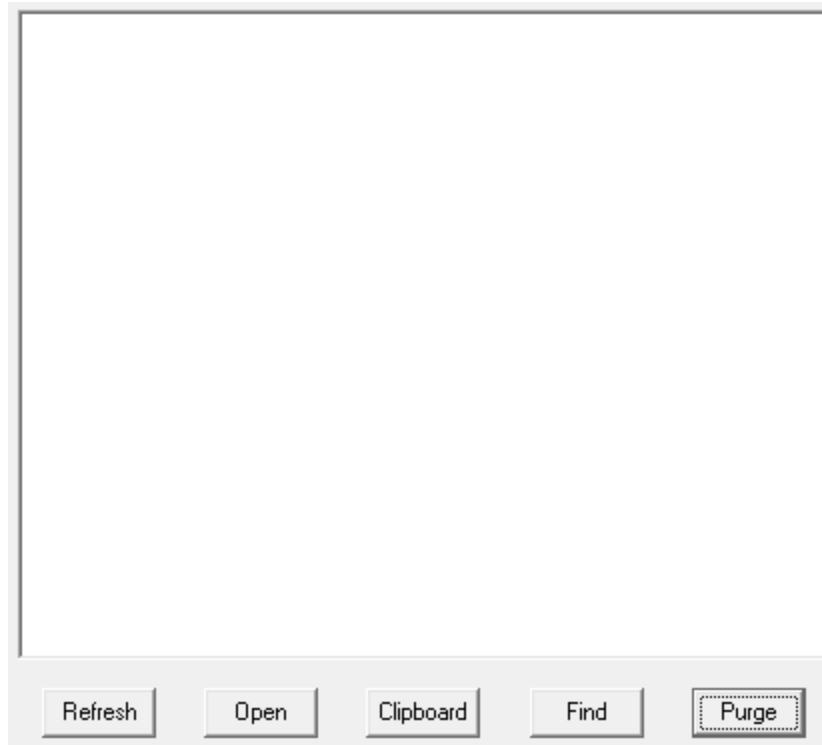
Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

PageGate Admin: How to Configure

6

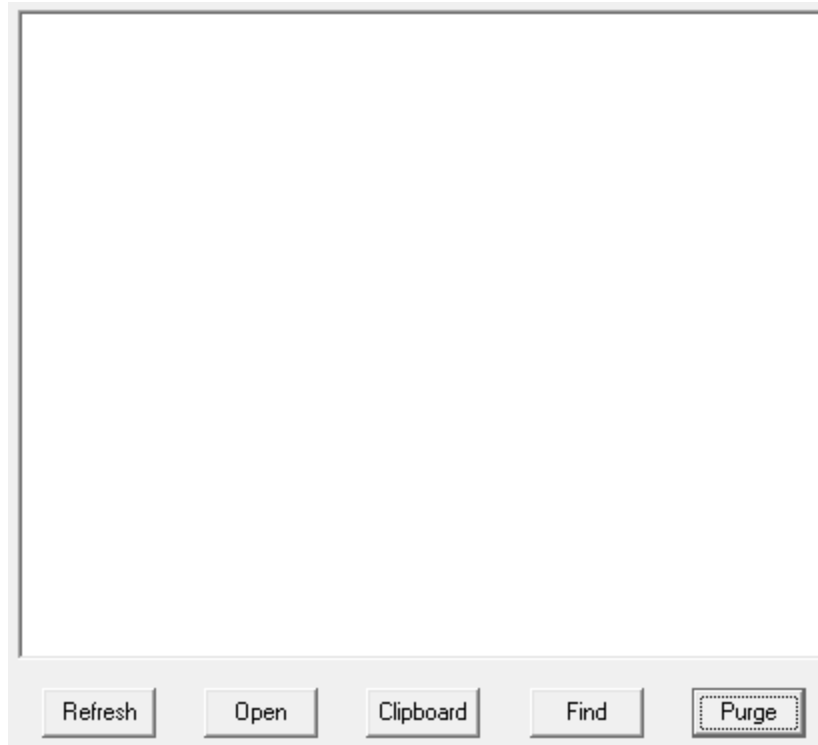
PageGate Scheduler

This log shows a summary of the events generated by the PageGate Scheduler.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

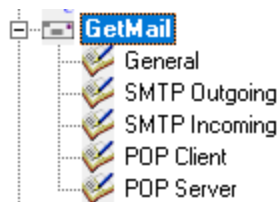
This log is a record of information and traffic processed by the GetASCII module.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

GetMail

This section contains a record of all functions performed by GetMail and the PageGate Mail Server service. As the GetMail module has many functions, it has five separate logs that correspond to it.



- [General](#)

PageGate Admin: How to Configure

PageGate

This log is a record of information and traffic processed by the GetMail module.

- [SMTP Outgoing](#)

This log is a record of all outgoing SMTP transactions negotiated by the PageGate Mail Server.

- [SMTP Incoming](#)

This log is a record of all incoming SMTP transactions negotiated by the PageGate Mail Server.

- [POP Client](#)

This log is a record of all POP mail collection sessions negotiated by the PageGate Mail Server.

- [POP Server](#)

This log is a record of all incoming POP mail collection requests negotiated by the PageGate Mail Server.

General

This log is a record of information and traffic processed by the GetMail module.



PageGate

Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

SMTP Outgoing

This log is a record of all outgoing SMTP transactions negotiated by the PageGate Mail Server.



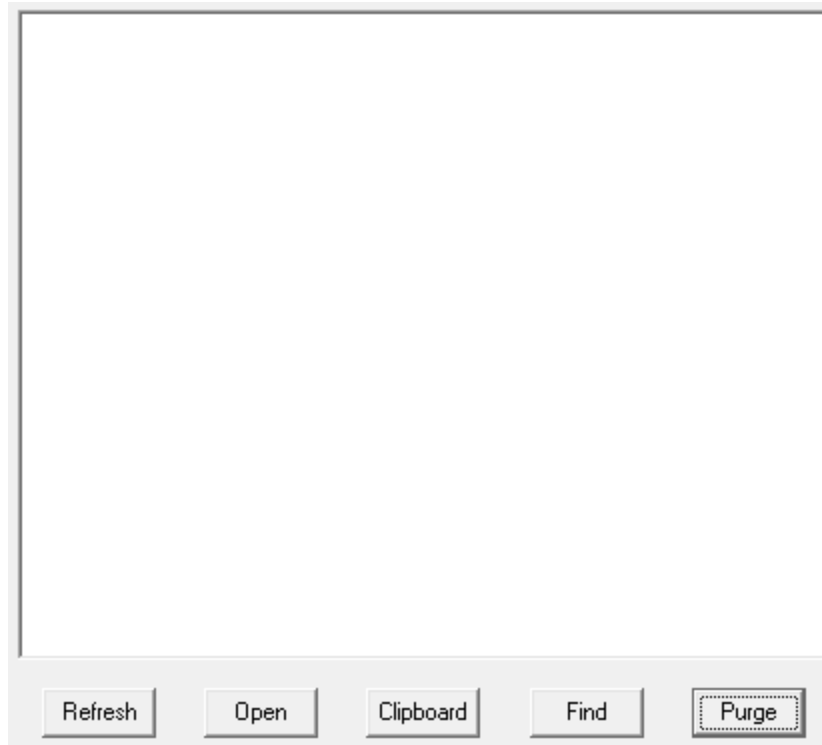
Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

PageGate Admin: How to Configure

6



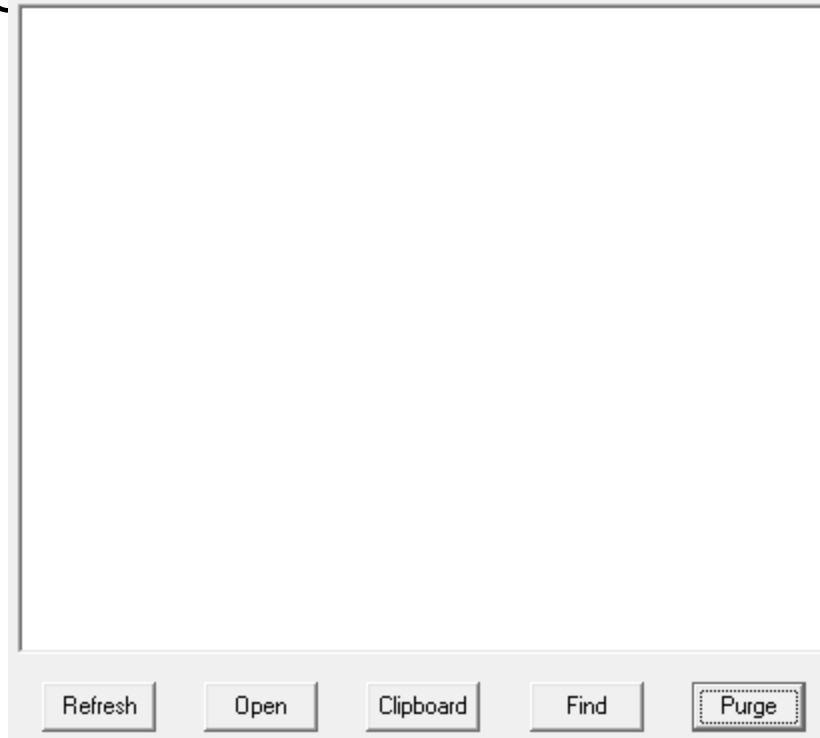
This log is a record of all incoming SMTP transactions negotiated by the PageGate Mail Server.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

POP Client

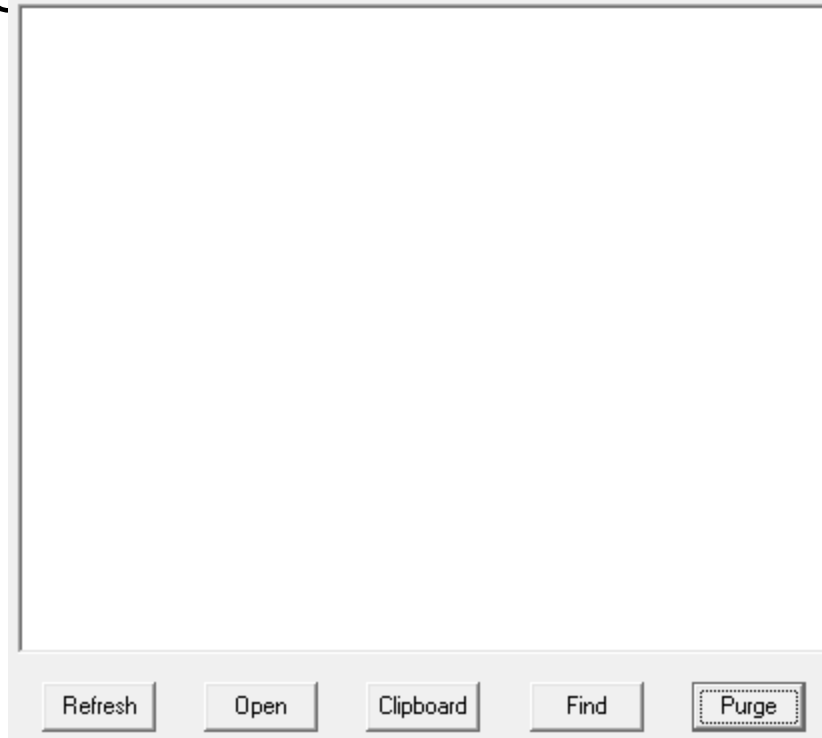
This log is a record of all POP mail collection sessions negotiated by the PageGate Mail Server.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

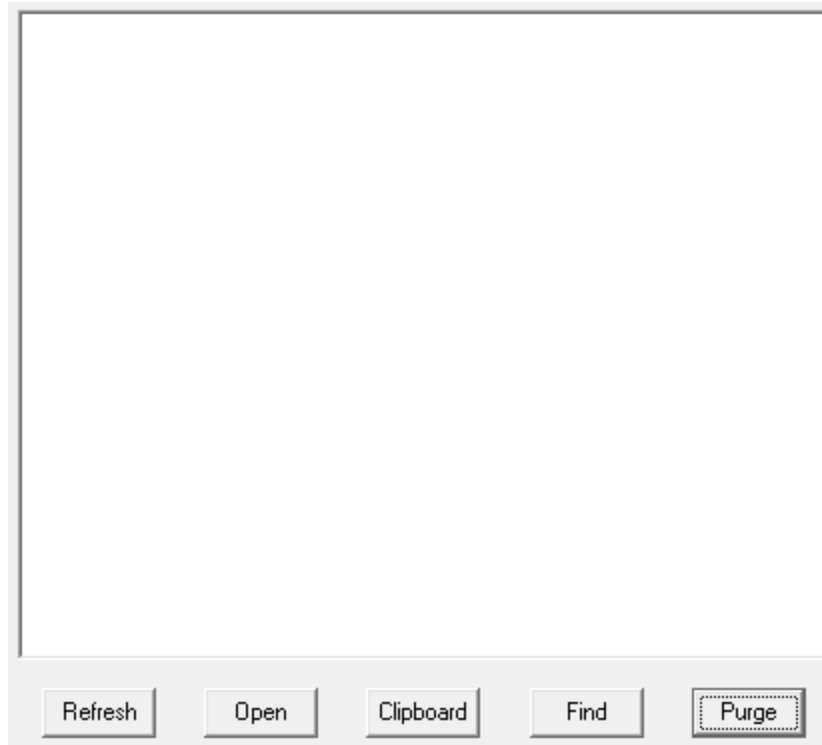
POP Server

This log is a record of all incoming POP mail collection requests negotiated by the PageGate Mail Server.



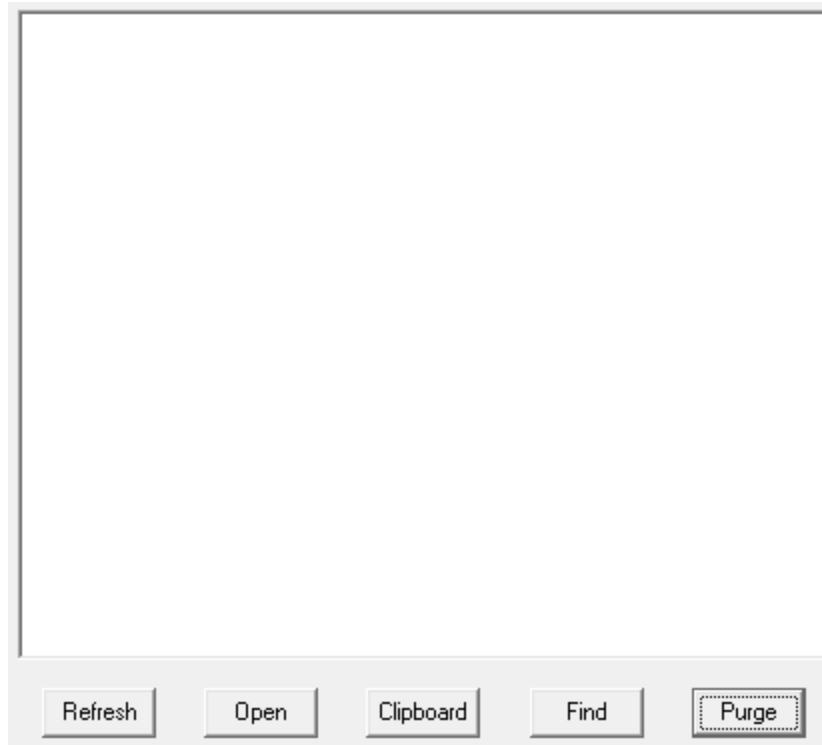
Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

This log is a record of information and traffic processed by the GetSerial module.



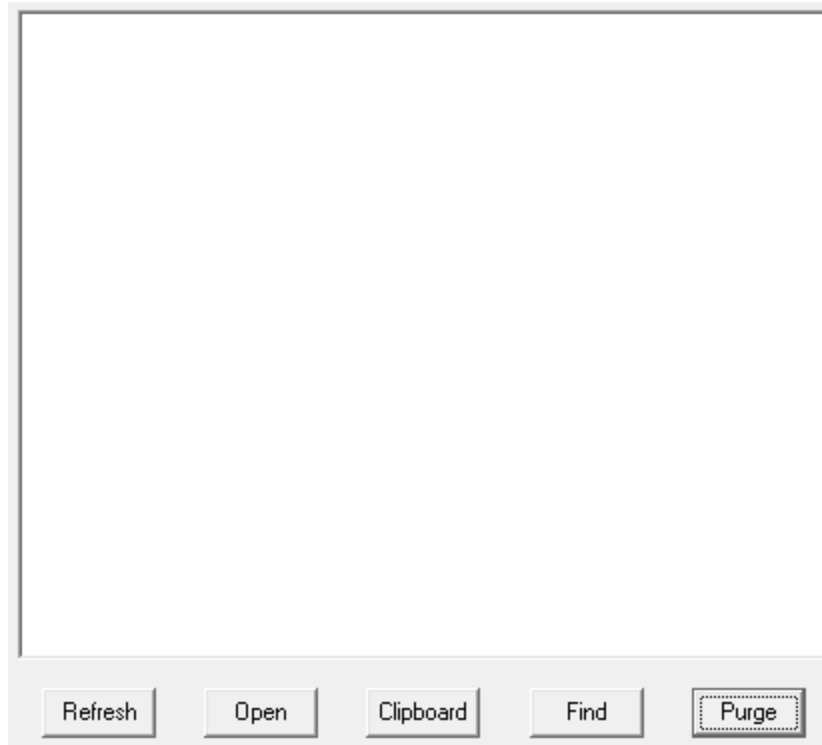
Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

This log is a record of information and traffic processed by the GetTap module.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

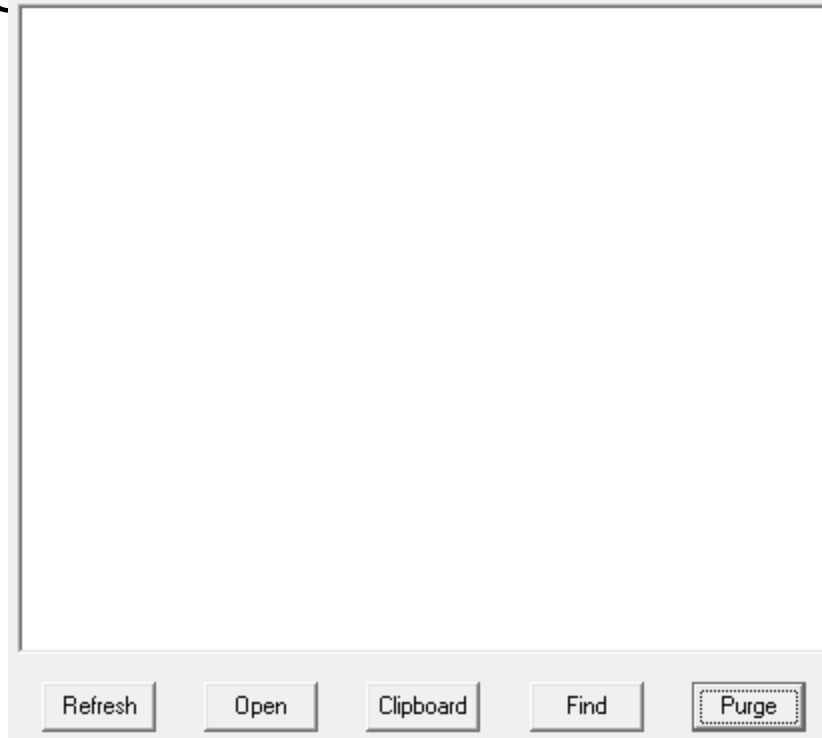
This log is a record of information and traffic processed by the GetWeb module.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

GUI Client

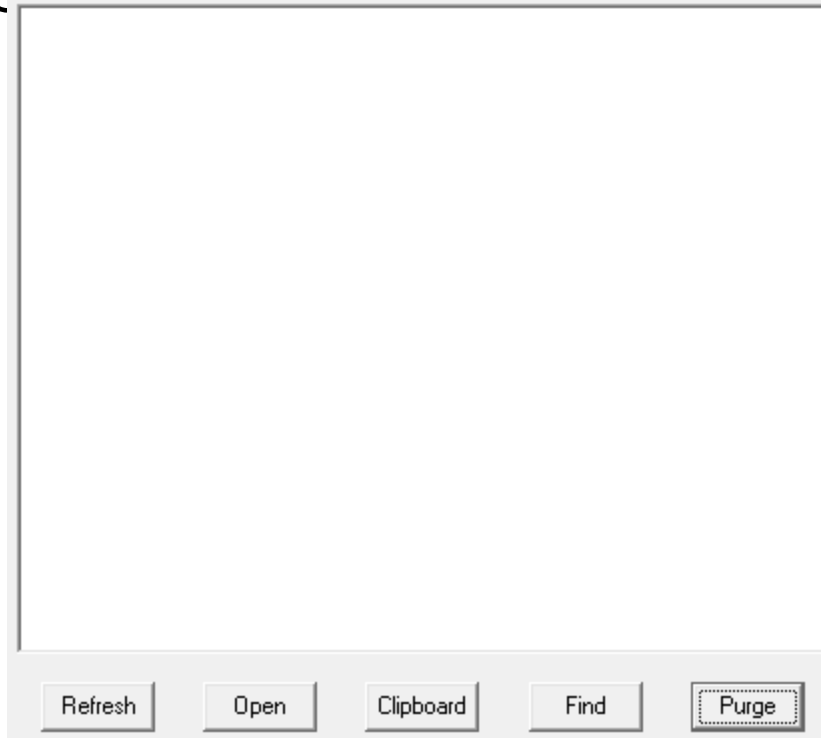
This log is a record of information and traffic submitted by workstations using the GUI Client.



Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

Connectors

The connector logs are a record of all protocol negotiations handled by the selected connector.



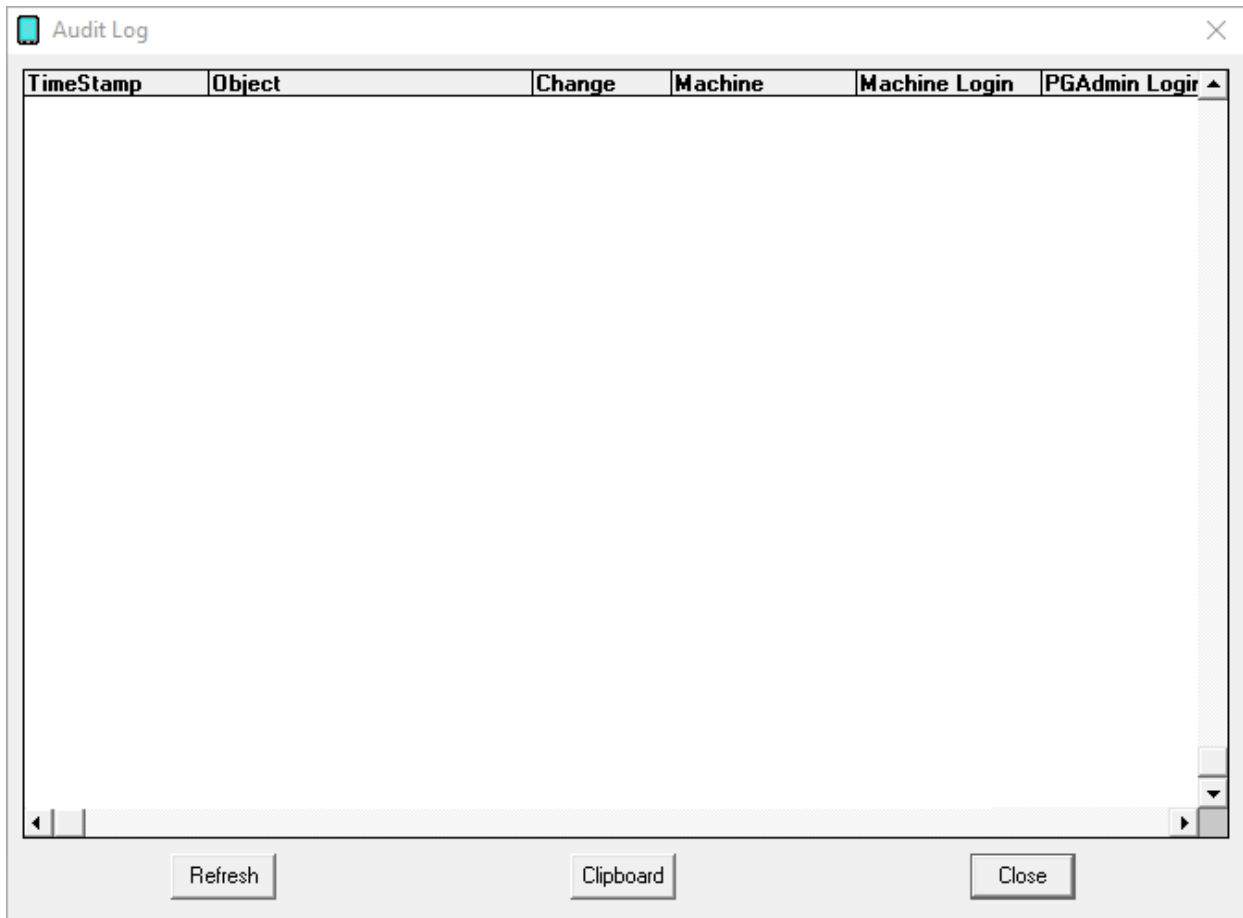
Button	Function
Refresh	Displays the most current entries in the log.
Open	Opens the selected log in the system default text editor.
Clipboard	Copies the contents of this log to the Windows Clipboard, allowing it to be pasted into another application.
Find	Allows you to search for something within the log.
Purge	Deletes the entirety of this log.

PageGate Admin: How to Configure

PageGate Audit

6

This is a record of every administrative function performed by any PageGate Admin.



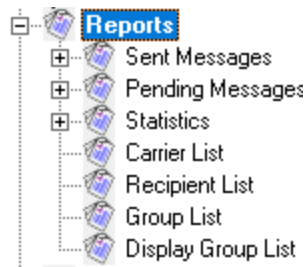
The audit log exists to provide a record of what users on what systems performed which actions in the PageGate Admin.

This log shows the date and time that an action took place, what object was interacted with, how the object was modified, what machine name the PG Admin was running on, what Windows user the machine was logged in with and what PG Admin credentials were being used.

Reports

This section allows you to run reports on sent and pending messages, statistical analysis of server performance and general lists of carrier, recipient or group information.

PageGate Admin: How to Configure PageGate



- [Sent Messages](#)

This section allows you to run reports on messages sent from PageGate by carrier, connector, interface, recipient, sender or GUI Client station ID or on all messages.

- [Pending Messages](#)

This section allows you to run reports on messages in the pending queue by carrier, interface, recipient, sender or GUI Client station ID or on all messages.

- [Statistics](#)

These reports allow you to see a statistical analysis of traffic processed by carrier, connector, interface, recipient, sender or GUI Client station ID.

- [Carrier List](#)

This report displays information about the configured carriers.

- [Recipient List](#)

This report displays information about configured recipients.

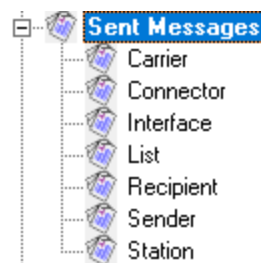
- [Group List](#)

This report displays information about configured groups.

- [Display Group List](#)

This report displays information about configured groups.

Sent Messages



- [Carrier](#)

This report will show what (and how many) messages have been sent through a selected carrier (or all carriers, if you select the All option).

PageGate Admin: How to Configure PageGate

- [Connector](#)

This report will show what (and how many) messages have been sent through a selected Connector (or all Connectors, if you select the All option).

- [Interface](#)

This report will show what (and how many) messages have been sent through a selected Interface (or all Interfaces, if you select the All option).

- [List](#)

This report will show all messages sent within a specified date and time range.

- [Recipient](#)

This report will show what (and how many) messages have been sent to a specific Recipient or Group.

- [Sender](#)

This report will show what (and how many) messages were sent by a specific user.

- [Station](#)

This report will show what (and how many) messages were sent by a specific GUI Client Station.

Carrier

Carrier(s)

[ALL]

ExampleCarrier1

Start Date End Date

Start Time End Time

Summary
 Details
 Totals Only

All Msgs
 Good Msgs
 Bad Msgs

Print report
 Zero Values

Preview
 Full Names

Write report to file

...

PageGate

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Connectors

<input type="checkbox"/> Connector1	<input type="checkbox"/> Connector2
<input type="checkbox"/> Connector3	<input type="checkbox"/> Connector4
<input type="checkbox"/> Connector5	<input type="checkbox"/> Connector6
<input type="checkbox"/> Connector7	<input type="checkbox"/> Connector8
<input type="checkbox"/> Connector9	<input type="checkbox"/> Connector10
<input type="checkbox"/> Connector11	<input type="checkbox"/> Connector12
<input type="checkbox"/> Connector13	<input type="checkbox"/> Connector14
<input type="checkbox"/> Connector15	<input type="checkbox"/> Connector16

Start Date End Date

Start Time End Time

Summary
 Details
 Totals Only

All Msgs
 Good Msgs
 Bad Msgs

Print report
 Zero Values

Preview
 Full Names

Write report to file

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.

All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Interface

Setting	Function
Start Date	The beginning date of the date range the report should show.

PageGate

End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.

PageGate Admin: How to Configure PageGate

6

Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Recipient

Recipient(s)

(ALL)

Start Date End Date

Start Time End Time

Summary
 Details
 Totals Only

All Msgs
 Good Msgs
 Bad Msgs

Print report
 Zero Values

Preview
 Full Names

Write report to file

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.

Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Sender

[All]

Start Date End Date

Start Time End Time

Summary
 Details
 Totals Only

All Msgs
 Good Msgs
 Bad Msgs

Print report
 Zero Values
 Preview
 Full Names

Write report to file
 ...

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.

All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Station

Station

[All]

Start Date End Date

Start Time End Time

Summary
 Details
 Totals Only

All Msgs
 Good Msgs
 Bad Msgs

Print report
 Zero Values
 Preview
 Full Names

Write report to file

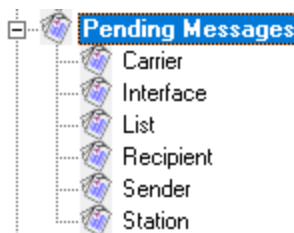
...

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.

PageGate

Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • Totals Only Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Pending Messages



- [Carrier](#)

PageGate Admin: How to Configure PageGate

This report will show what (and how many) messages are in the pending message queue for a selected carrier (or all carriers, if you select the All option).

- [Interface](#)

This report will show what (and how many) messages are in the pending message queue for a selected interface (or all interfaces, if you select the All option).

- [List](#)

This report will show all messages in the pending message queue for a specified date and time range.

- [Recipient](#)

This report will show what (and how many) messages are in the pending message queue for a specific recipient or group.

- [Sender](#)

This report will show what (and how many) messages in the pending message queue were sent by a specific user.

- [Station](#)

This report will show what (and how many) messages in the pending message queue were sent by a specific GUI Client Station.

Carrier

Carrier(s)

[ALL]

Repeating Messages Only

Summary
 Details
 Totals Only

Print report
 Zero Values

Preview
 Full Names

Write report to file

...

PageGate

Setting	Function
Repeating Messages Only	Enabling this option will only report on repeating messages scheduled by a GUI Client workstation.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Interface

Interfaces

Admin Client
 GetAscii GetMail
 GetSerial GetTap
 GetWeb

Repeating/Unscheduled Messages Only

Summary Details Totals Only

Print report Zero Values
 Preview Full Names

Write report to file

...

Setting	Function
---------	----------

PageGate

Repeating/Unscheduled Messages Only	Enabling this option will only report on repeating messages and messages scheduled to go out at a future date and time.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

List

The screenshot shows a configuration window with the following options:

- Repeating/Unscheduled Messages Only
- Summary Details Totals Only
- Print report
 - Preview
 - Full Names
- Write report to file
 - Text input field: _____
 - File selection button: ...
- Print button

Setting	Function
Repeating/Unscheduled Messages Only	Enabling this option will only report on repeating messages scheduled by a GUI Client workstation.

PageGate Admin: How to Configure PageGate

6

Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Recipient

Setting	Function
Repeating Messages Only	Enabling this option will only report on repeating messages scheduled by a GUI Client workstation.

Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Sender

Setting	Function
Repeating Messages Only	Enabling this option will only report on repeating messages scheduled by a GUI Client workstation.

PageGate Admin: How to Configure PageGate

6

Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

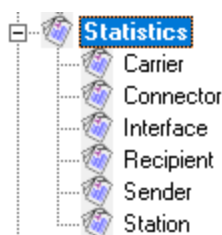
Station

Setting	Function
Repeating Messages Only	Enabling this option will only report on repeating messages scheduled by a GUI Client workstation.

Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Full Names	When this option is enabled, the Full Name field is used in lieu of the Recipient field in the reports.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Statistics

These reports measure rate volume of throughput, rate of delivery and total delivery results.



- [Carrier](#)

This report will show the statistical information for selected carrier (or all carriers, if you select the All option).

- [Connector](#)

This report will show the statistical information for a selected connector (or all connectors, if you select the All option).

- [Interface](#)

This report will show the statistical information for a selected interface (or all interfaces, if you select the All option).

PageGate Admin: How to Configure PageGate

- [Recipient](#)

This report will show the statistical information for a selected recipient or group (or all recipients and groups, if you select the All option).

- [Sender](#)

This report will show the statistical information for messages sent by a specific user.

- [Station](#)

This report will show the statistical information for messages sent by a specific GUI Client Station ID.

Carrier

Carrier(s)

(ALL)

Start Date

End Date

Start Time

End Time

All Msgs
 Good Msgs
 Bad Msgs

Print report

Preview

Write report to file

...

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.

All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Connector

Connectors

<input type="checkbox"/> Connector1	<input type="checkbox"/> Connector2
<input type="checkbox"/> Connector3	<input type="checkbox"/> Connector4
<input type="checkbox"/> Connector5	<input type="checkbox"/> Connector6
<input type="checkbox"/> Connector7	<input type="checkbox"/> Connector8
<input type="checkbox"/> Connector9	<input type="checkbox"/> Connector10
<input type="checkbox"/> Connector11	<input type="checkbox"/> Connector12
<input type="checkbox"/> Connector13	<input type="checkbox"/> Connector14
<input type="checkbox"/> Connector15	<input type="checkbox"/> Connector16

Start Date End Date

Start Time End Time

All Msgs
 Good Msgs
 Bad Msgs

Print report

- Preview

Write report to file

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.

PageGate

End Time	The beginning time of the time range the report should show.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Interface

Interfaces

Admin Client
 GetAscii GetMail
 GetSerial GetTap
 GetWeb

Start Date End Date
 Start Time End Time

All Msgs Good Msgs Bad Msgs

Print report
 Preview
 Write report to file

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.

PageGate

End Time	The beginning time of the time range the report should show.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Recipient

Recipient(s)

[ALL]

Start Date End Date

Start Time End Time

All Msgs
 Good Msgs
 Bad Msgs

Print report
 Full Names

Preview

Write report to file

...

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.

PageGate

End Time	The beginning time of the time range the report should show.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Sender

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.

PageGate Admin: How to Configure PageGate

All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Station

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.

PageGate Admin: How to Configure PageGate

6

All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Carrier List

This report displays information about the configured carriers. To run a report on all carriers, select the (ALL) option. To run a report on individual carriers, select the carrier in the list.

Setting	Function
Summary	Selecting this option will report the basic information (protocol and configured communication parameters) for the selected carrier(s).
Details	Selecting this option will report the all information (protocol, configured communication parameters, message limit, character limit, etc) for the

	selected carrier(s).
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Recipient	Enabling this option will tell the report to include a list of all recipients tied to the selected carrier(s).
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Recipient List

This report displays information about the configured recipients. To run a report on all recipients, select the (ALL) option. To run a report on individual recipients, select them in the list.

Setting	Function
Summary	Selecting this option will report the basic information (short name, full name, carrier, ID/PIN, failover, type and forwarding information) for the selected recipient(s).
Details	Selecting this option will report the all information (short name, full name, carrier, ID/PIN, failover, type, forwarding information, enabled services and content of comment fields) for the selected recipient(s).
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the

PageGate Admin: How to Configure PageGate

6

	preview, in which case nothing is printed, or print the report.
Full Names	Enabling this option will tell the report to include the recipients' full names when outputting the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Group List

This report displays information about the configured groups. To run a report on all groups, select the (ALL) option. To run a report on individual groups, select them in the list.

Setting	Function
Summary	Selecting this option will report the basic information (protocol and configured communication parameters) for the selected carrier(s).
Details	Selecting this option will report the all information (protocol, configured communication parameters, message limit, character limit, etc) for the selected carrier(s).
List by Group	Selecting this option will sort the output of the report based on group name.
List by Recipient	Selecting this option will sort the output of the report based on recipient name.
Show Empty	Enabling this option allows the report to display groups with no members.
Full Names	Enabling this option tells the report to output the reported groups' full names.

PageGate

Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Display Group List

This report displays information about the configured display groups. To run a report on all display groups, select the (ALL) option. To run a report on individual display groups, select them in the list.

Setting	Function
Summary	Selecting this option will report the basic information (protocol and configured communication parameters) for the selected carrier(s).
Details	Selecting this option will report the all information (protocol, configured communication parameters, message limit, character limit, etc) for the selected carrier(s).
List by Group	Selecting this option will sort the output of the report based on group name.
List by Recipient	Selecting this option will sort the output of the report based on recipient name.
Show Empty	Enabling this option allows the report to display groups with no members.

PageGate

Full Names	Enabling this option tells the report to output the reported groups' full names.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Advanced Server and Client Settings

There are several settings in PageGate that are not contained within the Administrative Interface. These settings are not commonly used but are available for knowledgeable Administrators to control certain advanced functions within the Server and Client.

All of these Advanced functions are controlled through Registry entries, so you **MUST** be logged in to Windows with an Administrative Account to configure them.

These registry settings are for modifying the default behavior of the program. Unless they are needed, these registry entries **SHOULD NOT EXIST**.

- [Registry settings for PageGate Server](#)
- [Registry settings for PageGate Client](#)

Registry settings for PageGate Server

There are several settings in PageGate that are not contained within the Administrative GUI Interface. These settings are not commonly used but are available for knowledgeable Administrators to control certain advanced functions within the Server. All of these Advanced functions are controlled through Registry entries, so you **MUST** be logged in to Windows with an Administrative Account to configure them.

These registry settings are for modifying the default behavior of the program. Unless they are needed, these registry entries **SHOULD NOT EXIST**.

These registry entries must be created in the HKEY_LOCAL_MACHINE - Software - Wow6432Node - NotePage - PageGate section of the registry.

Name	Type	Value	Function
DNS_Spam_List	String	See Function	This value is a comma separated list of DNS Servers that are used to check for blacklisted domains. For more detailed information, reference the DNSBL.

PageGate

Drop_Msgs_with_Exes	Dword	0 for False 1 for True	If enabled, PageGate's GetMail module will not process inbound email messages that have an executable attached to them.
GetAscii_Unhide	String	True False	By default, GetAscii cannot see hidden files. To enable it to see hidden files, set this registry value to True.
GetSerial_Ignore_Restart	String	True False	By default, GetSerial is restarted during the database maintenance routine at midnight. To prevent this from happening, set this registry setting to True.
GetTap_Ignore_Restart	String	True False	By default, GetTAP is restarted during the database maintenance routine at midnight. To prevent this from happening, set this registry setting to True.
GetTapIdExtras	String	Alphanumeric	This setting allows GetTAP to provide extra (non-standard) characters after the ID= response in the TAP delivery sequence. For example if you want GetTAP to send ID=<CR><LF> instead of the standard ID=, you would set this registry value to \013\010
Max_POP_Collection_Threads	Dword	0 - 50	This value sets the maximum number of concurrent outbound POP Collection threads. It is recommended that this value never be set higher than 50. Standard values for this setting are between 20 and 30.
Max_POP_Listener_Threads	Dword	0 - 50	This value sets the maximum number of concurrent inbound POP Listener threads. It is recommended that this value never be set higher than 50. Standard values for this setting are between 20 and 30.
Max_SMTP_Delivery_Threads	Dword	0 - 50	This value sets the maximum number of concurrent outbound SMTP Delivery threads. It is recommended that this value never be set higher than 50. Standard values for this setting are between 20 and 30.
Max_SMTP_Listener_Threads	Dword	0 - 50	This value sets the maximum number of concurrent inbound SMTP Listener threads. It is recommended that this value never be set higher than 50. Standard values for this setting are between 20 and 30.
No_Admin_Backup_Prompt	Dword	0 for False 1 for True	Settings this value to true prevents PG Admin from prompting users to make a backup of the PageGate database.
NoLoginOption	Dword	0 for False 1 for True	Settings this value to True prevents login prompts from being displayed by the PageGate Admin. If PageGate's security system is enabled, only users

			<p>0 No Authentication Required. NOTE: Unless required, do not use this value.</p> <p>1 No authentication is required if the recipient in the Rcpt To command is addressed to a domain configured in the GetMail Module and its Aliases file. Traffic addressed to any other domain must authenticate for messages to be accepted.</p> <p>2 All inbound SMTP connections must use authentication.</p>
SMTP_Delivery_Port	Dword	Numeric	This entry specifies the Port that PageGate's SMTP server should use to deliver SMTP messages. By default, PageGate uses port 25 for outgoing SMTP connections. If you need to specify a different port, do so with this entry.
SMTP_IP	String	IP Address	If the PageGate is installed on a server that is multi-homed (uses multiple network interface cards), by default, the PageGate Mail Server will listen for incoming SMTP connections on all NIC's and IP addresses. This setting forces the PageGate Mail Server process to only listen on the IP specified.
SMTP_Port	Dword	Numeric	This entry specifies the Port on which PageGate's SMTP server should respond to SMTP connection requests. By default, PageGate monitors port 25 for incoming SMTP connections. If you need to specify a different port, do so with this entry.
SMTP_Timeout	Dword	Milliseconds	This field is expressed in milliseconds and represents how long PageGate's SMTP listener should wait for a response from connecting systems.
Zip_Log_Archive	String	True False	By default, PageGate will zip each day's archived logs. To disable this behavior and have the program keep the uncompressed dated folder and files in the logging directory, set this to 0.

Registry settings for PageGate Client

If you have the PageGate Client configured to store settings on a per-profile basis, these registry settings will go in:

Computer\HKEY_CURRENT_USER\SOFTWARE\NotePage\NotePagerNet

PageGate Admin: How to Configure PageGate

6

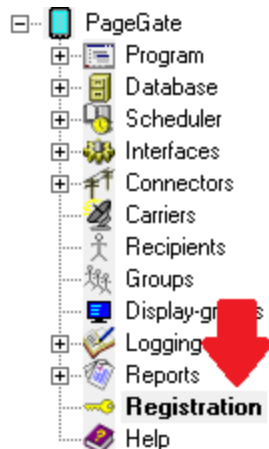
If you have the PageGate Client configured to store settings for all local users on a workstation, these registry entries will go in:

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\NotePage\NotePageNet

Name	Type	Value	Function
AllHistory			
NoLoginOption	Dword	0 for False 1 for True	Settings this value to True prevents login prompts from being displayed by the PageGate Client. If PageGate's security system is enabled, only users whose Windows accounts have been given login permission can open PG Client.
RecipientListFile			
SyncTimeWithServer			

Registration

This section allows you to enter your PageGate registration key.



Clicking on Registration will provide this dialog:

PageGate Admin: How to Configure PageGate

You can manually enter the registration key but to make things easier, we've included a 'Paste Registration From Clipboard' option.

In the email containing your registration key, you'll see a section like this:

```

--- Start of PageGate Registration Key Information ---
Registration Key Information (must be typed exactly as shown)
Name      : Name
Company Name : Company Name
Address1   : Address
Address2   :
City      : City
State     :State
Zip Code  :Zip Code
Country   : Country
  
```

Registration Key: 123456-789012-3456789-012345

```

--- End of PageGate Registration Key Information ---
  
```

If you highlight that section and copy it to your clipboard, you can click the 'Paste Registration From Clipboard' to have the key automatically inserted.

After entering your PageGate version 10 registration key, click Apply then close the Registration screen.

PageGate Admin: How to Configure PageGate

6

Not every method of delivery supports a reply route, so it isn't always possible for people and automated systems to reply to the messages and API calls you send. However, many methods of delivery do support reply systems and this gives you the ability to use two-way messaging with PageGate.

[Two Way Messaging with PageGate's GUI Client](#)

When using either of 2-way messaging methods listed below, replies can be sent back to workstations using the GUI Client. This allows conversations to be initiated by the PageGate Client and for replies to those messages to come back to the workstation. This feature also allows client workstations to message each other in a chat style interaction.

[HTTP/S response by API or Protocol](#)

This is most commonly used when delivering messages with [a messaging API](#) or [carrier enterprise system](#) like [AT&T's Firstnet](#) and [Verizon's EMAG](#) gateways. This is also used when bi-directionally communicating with ticket tracking systems, alarm systems and similar.

PageGate has an HTTP/S data receiver (webdata.exe) that can receive input and responses from nearly any API or protocol that offers an HTTP return like HTTP GET, HTTP POST and WCTP.

APIs and protocols that have the ability to send replies require that you provide a URL for the responses to go to. This means that, typically speaking, you will need to host PageGate's webdata receiver on a web server the host can respond to. In some cases, this will be an internet host and you will need to host PageGate's webdata receiver on an internet accessible web server. Otherwise, the hosts won't be able to route the replies and responses to you.

[Click here for more information and configuration steps.](#)

[Cellular Hardware](#)

When using [cellular hardware](#) to deliver SMS, the people who receive those SMS have the option to reply to them. When they reply, their message is delivered back to the sending cellular device, where it waits in memory. PageGate can be configured to check the cellular hardware for reply messages received by these devices and process them in a number of ways.

In the most basic configuration, all reply messages are sent to a [specific recipient or group](#). However, reply messages can be fed through the [GetASCII API for basic](#)

Two Way Messaging and Reply Processing

7

[processing](#), which allows you to conditionally modify what is done with the replies in some circumstances.

Using a [Filter Pack in conjunction with the SMS replies](#) and the GetASCII API allows you to modify any part of the replies being processed, from what the message should say to whom the message should be delivered. It also allows you to implement conditional modifications. For example, if a certain word or keyphrase appears in the body of the message, you could have the filter pack completely change who that message is intended to go to.

Two Way Messaging with PageGate's GUI Client

If a PageGate GUI Client's "Sender Name" value matches the name of a [local recipient](#) in PageGate, messages sent to that local recipient will appear in that workstation's GUI client 'Replies' window. This provides a way of sending messages between workstations and also provides the ability for GUI Client workstations to receive replies to delivered messages.

To configure two-way messaging for PageGate's GUI Client, you'll need to:

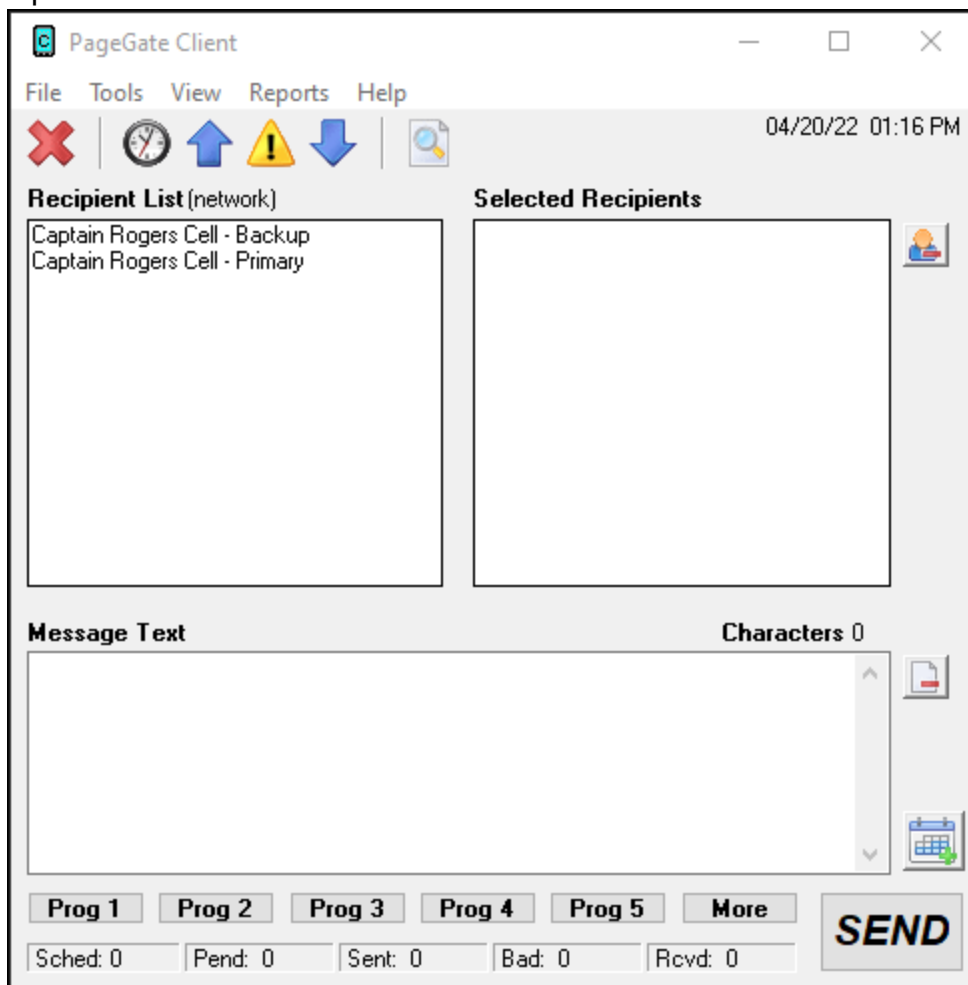
Step 1: Implement a method of two-way communication

Not every method of delivery supports a reply route, so it isn't always possible for people and automated systems to reply to the messages and API calls you send. However, many methods of delivery do support reply systems and this gives you the ability to use two-way messaging with PageGate.

Step 2: Configure the "Sender's Name" in each workstation that should be able to receive replies.

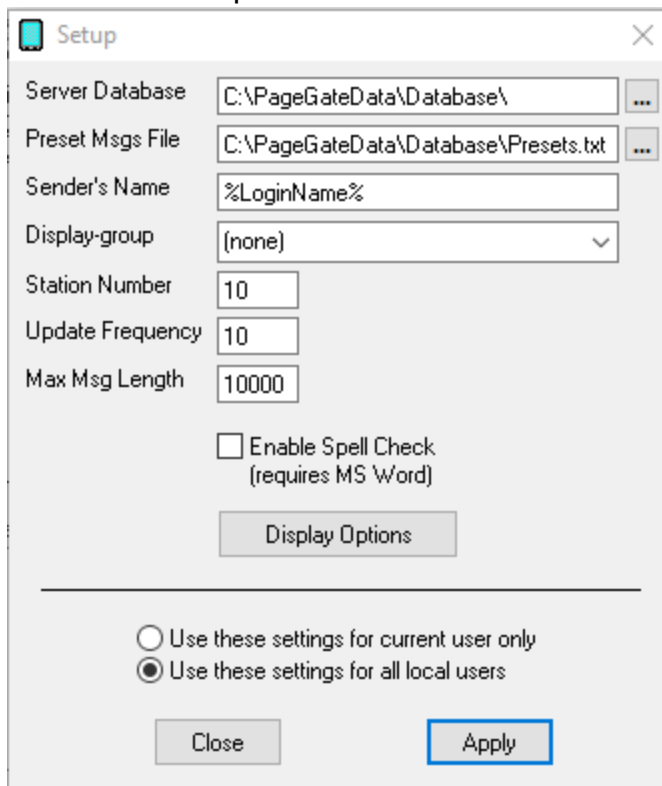
Two Way Messaging and Reply Processing

1) Open the PG GUI Client.



Two Way Messaging and Reply Processing

2) Go to File - Setup.

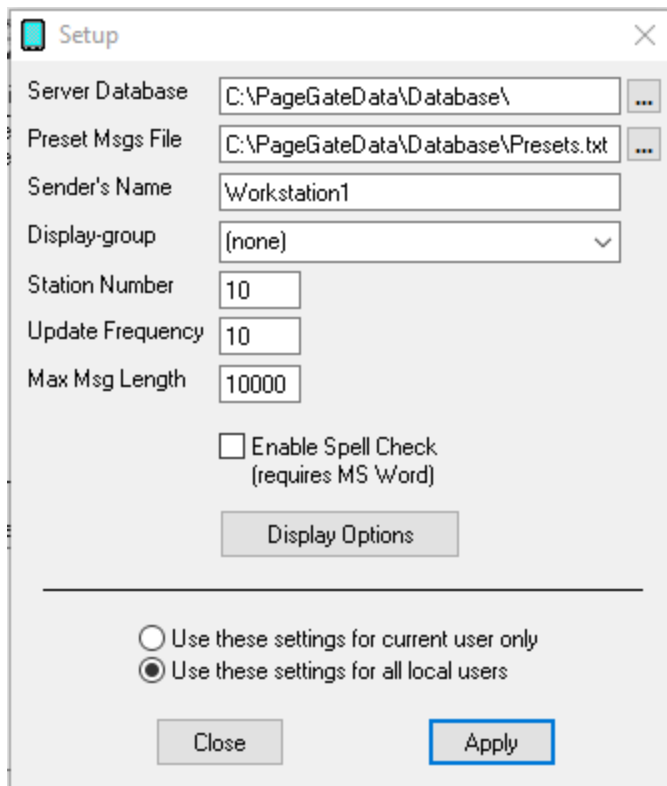


The Setup dialog box is shown with the following settings:

- Server Database: C:\PageGateData\Database\
- Preset Msgs File: C:\PageGateData\Database\Presets.txt
- Sender's Name: %LoginName%
- Display-group: (none)
- Station Number: 10
- Update Frequency: 10
- Max Msg Length: 10000
- Enable Spell Check (requires MS Word)
- Use these settings for current user only
- Use these settings for all local users

Buttons: Close, Apply

3) Set the "Sender's Name"



The Setup dialog box is shown with the following settings:

- Server Database: C:\PageGateData\Database\
- Preset Msgs File: C:\PageGateData\Database\Presets.txt
- Sender's Name: Workstation1
- Display-group: (none)
- Station Number: 10
- Update Frequency: 10
- Max Msg Length: 10000
- Enable Spell Check (requires MS Word)
- Use these settings for current user only
- Use these settings for all local users

Buttons: Close, Apply

Processing

4) Click Apply.

Step 3: Create local recipients

You'll need to create a local recipient whose name exactly matches a value configured as a sender name in the GUI Client. [Click here for steps on how to create local recipients.](#)

Step 4: Configure your reply processing system

The final piece is to tell your reply processing system to start referencing the local recipients you've created for the workstations. This can vary wildly depending on how you deliver messages and receive replies to them and we would recommend reviewing the reply processing steps associated with your implemented method.

[Click here for information on processing replies from APIs and protocols.](#)

[Click here for more information on processing replies from cellular hardware](#)

Replies with Cellular Hardware

When using cellular hardware to deliver text messages, the people who receive those SMS have the option to reply to them. When they reply, their reply message is actually delivered back to the sending cellular device, where it waits in memory. PageGate can be configured to check the cellular hardware for reply messages received by these devices.

Before you can configure reply processing, you'll first need to go in to the [2-Way sub-section](#) of a connector configured to use cellular hardware and enable the 'GSM-AT Replies' function. Once you do, you'll need to specify a 'Reply Recipient' that should be used to process the replies. This allows you to implement reply processing in a few ways.

- [Static Recipient or Group](#)

This is the most basic form of reply processing. All replies will be delivered to the specified recipient or group.

When used in conjunction with [two-way messaging in PageGate's GUI Client](#), this allows you to route replies to a workstation or group of workstations and also allows those workstations to reply back.

- [Reply processing with the GetASCII API](#)

If you own the GetASCII API, you can feed the reply information through that system for advanced processing. This allows you to modify message content and who the replies should go to but does not allow for conditional modification or conditional sending.

For example, you couldn't have the program scan the body of the message for a keyphrase, then completely change who the message is intended to go to based on it. That would require the [PageGate Filter Pack](#).

- [Reply processing with the GetASCII API and the PageGate Filter Pack](#)

Two Way Messaging and Reply

7

Processing

Reply messages can be fed through PageGate's [Filter Pack](#) to allow for conditional sending. This is required to sort replies to specific workstations when using [two-way messaging with PageGate's GUI Client](#).

For example, with the PageGate's filter pack, you can tell PageGate to conditionally change who receives the message based on what the text in the reply is. As an example, you could configure a filter system so that any time the word lunch shows up in a reply message, that reply is sent to the cafeteria group instead of the general replies group.

You could configure a system where replies from a certain phone number will always be copied to certain recipients or groups.

You could configure a system where certain keywords or phrases are replaced with other keywords or phrases. For example, you could translate a code like PD4N in to 4th Precinct.

You could configure a system to describe every element I've just described all at once.

However, you must own and implement the [Filter Pack](#) to have the functionality necessary to do conditional sending as described.

Static Recipient or Group

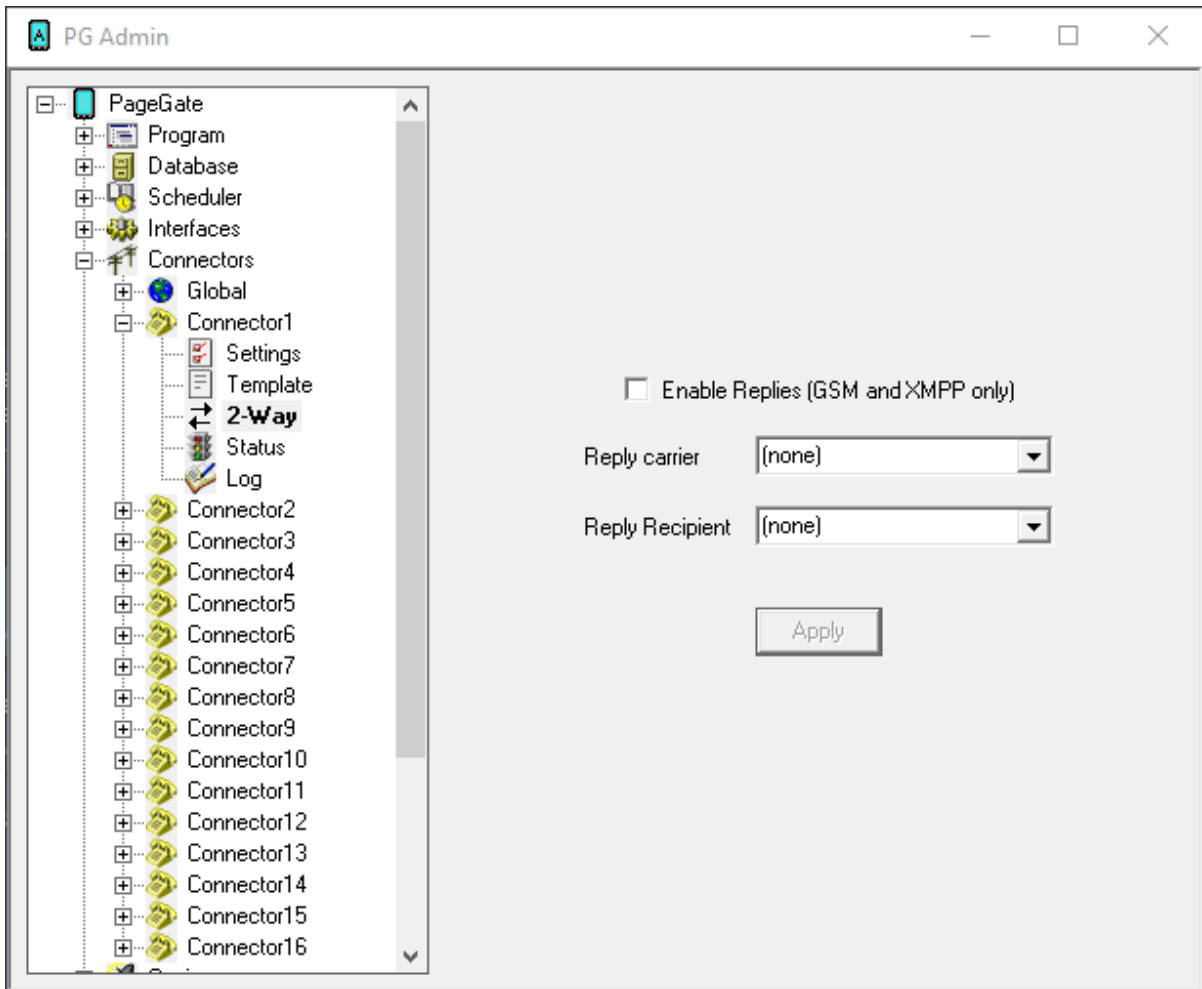
These are the steps on how to configure PageGate to deliver SMS replies from cellular hardware to a static recipient or group. When used in conjunction with [two-way messaging in PageGate's GUI Client](#), this allows you to route replies to a workstation or group of workstations and also allows those workstations to reply back.

Note: This only applies to replies to messages sent from the [cellular hardware](#) this connector is configured to use.

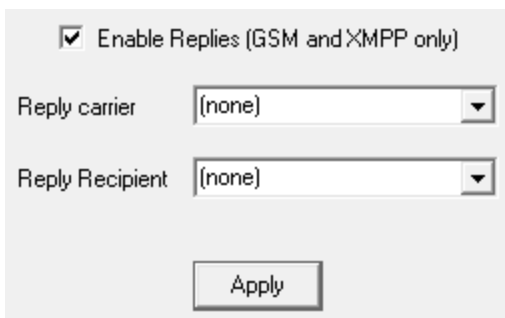
Two Way Messaging and Reply Processing

7

1) In the PageGate Admin, go to Connectors - Connector X - 2-Way.



2) Check 'Enable Replies (GSM-AT and XMPP only).'



Two Way Messaging and Reply Processing

- 3) Set 'Reply Carrier' to the carrier using the cellular hardware.

Enable Replies (GSM and XMPP only)
 Reply carrier: CellGateway1
 Reply Recipient: (none)
 Apply

- 4) Set 'Reply Recipient' to the recipient or group reply messages should route to.

Enable Replies (GSM and XMPP only)
 Reply carrier: CellGateway1
 Reply Recipient: reply_group
 Apply

- 5) Click Apply.

Processing replies with GetAscii

This is an example of how reply messages can be processed with the GetASCII API.

- 1) In Windows, create a new directory for PageGate's message processing to reference. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource.

However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation.

We recommend creating a directory called Replies as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\Replies\).

- 2) [Create a File protocol](#) carrier and set its file path to the folder you created in step 1. Leave the extension at .txt and set Max Chars to 6000.
- 3) [Create a recipient](#) and tie it to the File protocol carrier. Set its Full Name field to Replies and set its Recipient field to replies.
- 4) Go in to the Template sub-section of your new recipient.
- 5) Copy and paste the following in the Text section:

Two Way Messaging and Reply Processing

```
sms_replies_group
%Sender%
FROM: %Sender%
WHEN: %Date% %Time%
Message: %Message%
```

- 6) [Create a new group](#). Set its Description field to SMS Replies Group and its Group field to sms_replies_group
 - 7) Populate the group with members that should receive reply messages. If you include [local recipients](#), PageGate GUI Client workstations will receive replies as well.
 - 8) Go to Connectors - Connector X - 2-Way.
 - 9) Check "Enabled GSM-AT replies"
 - 10) Set the 'Reply Carrier' to the carrier using the cellular hardware.
 - 11) Set the 'Reply Recipient' to: replies
 - 12) Create a new folder on the local hard drive for the GetASCII module's basic polling directory. We recommend creating: C:\PageGateData\ASCII
- Note: If you have already configured a basic polling directory, you do not need to do steps 9, 10 or 11.
- 13) Go to [Interfaces - GetASCII - Settings](#)
 - 14) Set the Polling Directory to: c:\PageGateData\ASCII
 - 15) If unchecked, check Enabled, then click on Apply.
 - 16) Go to [Interfaces - GetASCII - Settings - Advanced Polling - Files](#)
 - 17) Add a new entry with the following settings:

```
Recipient: *
From:
File(s): *.txt
Path: Specify the folder you created in step 1, recommended c:\PageGateData\Replies\
File Type: Standard
```

- 18) Click Apply.

By configuring this example of a 2-way system, all replies will be read in and delivered to the replies recipient, which will output a file formatted to the template's specifications. The GetASCII API will then read in the re-formatted reply and send it to the recipients that are members of the group named sms_replies_group.

Processing

Processing replies with a Filter Pack

Note: PageGate's [filter pack](#) supports [RegEx expressions](#), which can allow you to implement an incredibly intricate system.

To say that there are many things you can do with a properly coded [filter pack](#) would be an understatement. The [Filter Pack](#) allows you to modify and conditionally modify any part of what it interacts with. It can reassign who a message is delivered to based on words or phrases in a reply, it can modify the body of the message, it can modify the subject line, it can insert pre-determined text and many other things.

For instance, with the filter pack, you can tell PageGate to conditionally change who receives the message based on what the text in the reply is. As an example, you could configure a filter system so that any time the word lunch shows up in a reply message, that reply is sent to the cafeteria group instead of the general replies group.

You could configure a system where replies from a certain phone number will always be copied to certain recipients or groups.

You could configure a system where certain keywords or phrases are replaced with other keywords or phrases. For example, you could translate a code like PD4N in to 4th Precinct.

You could configure a system to describe every element I've just described all at once, it's just a matter of coding your [filter pack](#) to perform the actions you want.

This section is designed to give you an idea of what a [filter pack](#) is capable of providing but does not represent its total capabilities.

[Example 1](#)

In this example, we'll configure a reply system to that any time someone replies to a message with 'unsubscribe' or 'subscribe', those replies are sent to an 'unsubscribe' or 'subscribe' distribution group for notification. If 'subscribe' or 'unsubscribe' aren't found, the reply is sent to a general replies recipient or group.

[Example 2](#)

In this example, we'll configure a reply system to that any time someone replies to a message with 'unsubscribe' or 'subscribe', those replies are sent to a recipient named 'unsubscribe' or 'subscribe' that executes a batch file or script. If 'subscribe' or 'unsubscribe' aren't found, the reply is sent to a general replies recipient or group. This can be especially useful if you're using an SQL environment and have PageGate's database linked to it with ODBC expressions. In theory, you could code a script to be executed that would automatically remove a responding phone number from the users table of the database any time the 'unsubscribe' script was executed.

Two Way Messaging and Reply

7

Processing

Example 1: Scanning for Keywords

In this example, we'll be going over how to have a reply processing system scan replies for keywords and keyphrases, like the words 'subscribe' and 'unsubscribe'.

Step 1: [Configure PageGate to send SMS with cellular hardware](#)

Step 2: [Configure PageGate to send Email](#)

In this particular configuration, we're going to have PageGate check replies for the words 'subscribe' or 'unsubscribe'. If it finds 'subscribe', PageGate is going to email that reply to the 'subscribe' recipient or group. If it finds 'unsubscribe', PageGate is going to email that reply to the 'unsubscribe' recipient or group.

Step 3: [Create a File carrier to pass the replies to PageGate's GetASCII API and Filter Pack for processing](#)

- 1) In Windows, create a new directory for PageGate's message processing to reference. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called Replies as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\Replies\).
- 2) Right click Carriers, left click Add.
- 3) Specify the following:
 - Carrier:** reply-processing
 - Protocol:** File
 - File Path:** Specify the directory you created in step 1.
By default, c:\PageGateData\Replies\
 - Extension:** .txt
 - Max Chars:** 2000
- 4) Click Apply.
- 5) Go to Carriers - reply-processing - Template.
- 6) In the Text section of the template, specify the following:
GeneralReplies
%Sender%

Two Way Messaging and Reply Processing

Message. %Message%

7) Click Apply.

Step 4: Create a recipient tied to the File carrier

1) Right click Recipients, left click Add.

2) Specify the following:

Full Name: message-replies

Recipient: message-replies

Type: Normal

Carrier: reply-processing

Failover: (none)

ID/PIN: 123

Max Chars: 2000

3) Click Apply

Step 5: Create a Recipient or Group named GeneralReplies

If the words 'subscribe' or 'unsubscribe' do not appear in a reply message, the reply will be sent to a recipient or group named GeneralReplies. For example, you could configure a series of 5 recipients tied to 5 separate email addresses, then add each recipient as a member of a group called GeneralReplies. When a reply needs to go to GeneralReplies, PageGate would then send that as an email to the 5 email recipients in the group. Alternatively, if GeneralReplies exists as a recipient, PageGate will send those messages to the email address or device configured for that recipient.

Step 6: Set the message-replies recipient as the Reply Recipient

1) Go to Connectors - Connector X - 2-way

Note: This only applies to replies to messages sent from the [cellular hardware](#) this connector is configured to use.

2) Check 'Enable Replies'

3) In 'Reply Carrier', select the carrier using the cellular hardware.

4) In 'Reply Recipient', select: message-replies.

5) Click Apply.

Step 7: Configure and enable the GetASCII API

Two Way Messaging and Reply Processing

7

- 1) In Windows, create a new directory for the GetASCII API's basic processing. We recommend creating a directory called ASCII as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\ASCII).
- 2) In the PageGate Admin, go to Interfaces - GetASCII - Settings.
- 3) Set the 'Polling Directory' to the directory you created in step 1. By default, c:\PageGateData\ASCII
- 4) Check Enabled.
- 5) Click Apply.
- 6) Say yes, you do want this to be enabled for everyone.
- 7) Go to Interfaces - GetASCII - Settings - Advanced Polling - Files.
- 8) Click Add and specify the following:
Recipient: *
From:
File(s): *.txt
Path: Specify the folder the reply messages will be written to.
By default, C:\PageGateData\Replies\
File Type: Standard
- 9) Click Apply.
- 10) Click Apply again.
- 11) Go to Program - Settings.
- 12) In the 'Run on this server' section, check GetASCII.
- 13) Click Apply.
- 14) If prompted for credentials, leave them blank and click Apply again.

Step 8: Configure the Filter Pack to process the replies based on 'subscribe' or 'unsubscribe' being in the reply message

- 1) Copy pgfilter.exe, pgfilter.xml and preproc.bat in to the directory you created for the reply-processing File carrier. By default, c:\PageGateData\Replies\
2) [Click here](#) for the necessary pgfilter.xml script configuration.

Once you finish this configuration, any time someone sends a reply to a message delivered by your [cellular hardware](#), PageGate will scan the body of the reply for 'subscribe' or 'unsubscribe'.

Two Way Messaging and Reply Processing

7

If it finds 'subscribe', it will send the reply to a specified recipient or group for delivery. If it finds 'unsubscribe', it will send the reply to a separate specified recipient or group for delivery.

Example 2: Changing Who the Message Goes To

In this example, we'll be going over how to have a reply processing system scan replies for keywords and keyphrases, like the words 'subscribe' and 'unsubscribe'. Once it finds them, it changes who the reply goes to and determines what sort of trigger happens.

Step 1: [Configure PageGate to send SMS with cellular hardware](#)

Step 2: [Create a File carrier to pass the replies to PageGate's GetASCII API and Filter Pack for processing](#)

1) In Windows, create a new directory for PageGate's message processing to reference. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called Replies as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\Replies\).

2) Right click Carriers, left click Add.

3) Specify the following:

Carrier: reply-processing

Protocol: File

File Path: Specify the directory you created in step 1.

By default, c:\PageGateData\Replies\

Extension: .txt

Max Chars: 2000

4) Click Apply.

5) Go to Carriers - reply-processing - Template.

6) In the Text section of the template, specify the following:

GeneralReplies

%Sender%

Message: %Message%

Two Way Messaging and Reply Processing

7

7) Click Apply.

Step 3: Create a recipient tied to the File carrier

1) Right click Recipients, left click Add.

2) Specify the following:

Full Name: message-replies

Recipient: message-replies

Type: Normal

Carrier: reply-processing

Failover: (none)

ID/PIN: 123

Max Chars: 2000

3) Click Apply

Step 5: Create a Recipient or Group named GeneralReplies

If the words 'subscribe' or 'unsubscribe' do not appear in a reply message, the reply will be sent to a recipient or group named GeneralReplies.

For example, you could configure a series of 5 recipients tied to 5 separate email addresses, then add each recipient as a member of a group called GeneralReplies. When a reply needs to go to GeneralReplies, PageGate would then send that as an email to the 5 email recipients in the group. Alternatively, if GeneralReplies exists as a recipient, PageGate will send those messages to the email address or device configured for that recipient.

Step 4: Set the message-replies recipient as the Reply Recipient

1) Go to Connectors - Connector X - 2-way

Note: This only applies to replies to messages sent from the [cellular hardware](#) this connector is configured to use.

2) Check 'Enable GSM-AT Replies'

3) In the Reply Recipient field, select: message-replies.

4) Click Apply.

Processing

Step 5: Configure a File carrier and recipient for the 'subscribe' action to execute a script or program

1) In Windows, create a new directory for PageGate's message processing to reference. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called subscribe-action as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\subscribe-action\).

2) Right click Carriers, left click Add.

3) Specify the following:

Carrier: subscribe-action

Protocol: File

File Path: Specify the directory you created in step 1.

By default, c:\PageGateData\subscribe-action\

Extension: .txt

Max Chars: 2000

4) Click Apply.

5) Right click Recipients, left click Add.

6) Specify the following:

Full Name: subscribe-recipient

Recipient: subscribe-recipient

Type: Normal

Carrier: subscribe-action

Failover: (none)

ID/PIN: Leave Blank

Max Chars: 2000

7) Click Apply.

Step 6: Configure a File carrier and recipient for the 'unsubscribe' action to execute a script or program

1) In Windows, create a new directory for PageGate's message processing to reference. This can be any folder accessible to PageGate, whether on the local hard drive or by UNC path to a network resource. However, if PageGate will need to reference a

Two Way Messaging and Reply Processing

7

network resource, the services relevant to these processes will need permission to access it. For more information, [see the services section](#) of the documentation. We recommend creating a directory called unsubscribe-action as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\unsubscribe-action\).

2) Right click Carriers, left click Add.

3) Specify the following:

Carrier: unsubscribe-action

Protocol: File

File Path: Specify the directory you created in step 1.

By default, c:\PageGateData\unsubscribe-action\

Extension: .txt

Max Chars: 2000

4) Click Apply.

5) Right click Recipients, left click Add.

6) Specify the following:

Full Name: unsubscribe-recipient

Recipient: unsubscribe-recipient

Type: Normal

Carrier: unsubscribe-action

Failover: (none)

ID/PIN: Leave Blank

Max Chars: 2000

7) Click Apply.

Step 7: Configure and enable the GetASCII API

1) In Windows, create a new directory for the GetASCII API's basic processing. We recommend creating a directory called ASCII as a sub-directory of [PageGate's database directory](#) (By default, c:\PageGateData\ASCII).

2) In the PageGate Admin, go to Interfaces - GetASCII - Settings.

3) Set the 'Polling Directory' to the directory you created in step 1. By default, c:\PageGateData\ASCII

4) Check Enabled.

5) Click Apply.

Two Way Messaging and Reply Processing

- 6) Say yes, you do want this to be enabled for everyone.
- 7) Go to Interfaces - GetASCII - Settings - Advanced Polling - Files.
- 8) Click Add and specify the following:
 - Recipient:** *
 - From:**
 - File(s):** *.txt
 - Path:** Specify the folder the reply messages will be written to.
By default, C:\PageGateData\Replies\
File Type: Standard
- 9) Click Apply.
- 10) Click Add and specify the following:
 - Recipient:** *
 - From:**
 - File(s):** *.txt
 - Path:** Specify the folder for the subscribe action.
By default, C:\PageGateData\subscribe-action\
File Type: Standard
- 11) Click Apply.
- 12) Click Add and specify the following:
 - Recipient:** *
 - From:**
 - File(s):** *.txt
 - Path:** Specify the folder the unsubscribe action.
By default, C:\PageGateData\unsubscribe-action\
File Type: Standard
- 13) Click Apply.
- 14) Click Apply again.
- 15) Go to Program - Settings.
- 16) In the 'Run on this server' section, check GetASCII.
- 17) Click Apply.
- 18) If prompted for credentials, leave them blank and click Apply again.

Processing

Step 8. Configure the Filter Pack in the Replies, subscribe-action and unsubscribe-action folders

- 1) Copy pgfilter.exe, pgfilter.xml and preproc.bat in to the directory you created for the reply-processing File carrier. By default, c:\PageGateData\Replies\
- 2) [Click here](#) for the necessary pgfilter.xml script configuration.
- 3) Copy preproc.bat in to the subscribe-action folder. By default, c:\PageGateData\subscribe-action.
- 4) Edit preproc.bat to execute whatever script and/or executable is appropriate for the function you want to perform.
- 5) Copy preproc.bat in to the unsubscribe-action folder. By default, c:\PageGateData\unsubscribe-action.
- 6) Edit preproc.bat to execute whatever script and/or executable is appropriate for the function you want to perform.

Once you finish this configuration, any time someone sends a reply to a message delivered by your [cellular hardware](#), PageGate will scan the body of the reply for 'subscribe' or 'unsubscribe'. If it finds 'subscribe', it will trigger whatever script has been configured in the preproc.bat file contained within the subscribe-action folder. If it finds 'unsubscribe', it will trigger whatever script has been configured in the preproc.bat contained within the unsubscribe-action folder.

Replies with APIs and Protocols

PageGate has an HTTP data receiver (webdata.exe) that can receive input from nearly any API or protocol that offers an HTTP return like HTTP GET, HTTP POST and WCTP. This allows you to receive JSON, XML, HTML, WCTP and similar data submissions via HTTP/S request.

Received data doesn't need to conform to a single standard as it is fully processed with the [GetASCII interface](#) and, optionally depending on what you want to do with the received data, [Filter Pack](#). If you want to receive multiple data formats, the [Filter Pack](#) isn't optional and you will need to implement a [filtering system](#) to tell PageGate what to do with each format received.

APIs and protocols that have the ability to send replies require that you provide a URL for the responses to go to. This means that, typically speaking, you will need to host PageGate's webdata.exe on an outward facing web server. Otherwise, the hosts won't be able to route the replies and responses to you.

webdata.exe is located in the PageGate program directory and MUST be copied to the webserver. If PageGate is not running on the web server hosting webdata.exe, you'll also

Two Way Messaging and Reply Processing

7

need to share the directory on the webserver where webdata.exe is hosted to allow the GetASCII interface to monitor replies sent to the receiver.

Before you begin, you'll need to determine what you want to do with the replies.

Do you want to auto-respond to the person who sent the message to let them know that they shouldn't reply?

Do you want to scan the reply for keywords and keyphrases to determine what to do?

What you want to do will determine which parts of the received data need to be keyed on. Since each API has its own potentially unique structure, you'll need to know which flags, tags and fields are relevant.

Reply messages can also be fed through PageGate's [Filter Pack](#) to allow for conditional sending. For instance, with the PageGate's filter pack, you can tell PageGate to conditionally change who receives the message based on what the text in the reply is. As an example, you could configure a filter system so that any time the word lunch shows up in a reply message, that reply is sent to the cafeteria group instead of the general replies group.

You could configure a system where replies from a certain phone number will always be copied to certain recipients or groups.

You could configure a system where certain keywords or phrases are replaced with other keywords or phrases. For example, you could translate a code like PD4N in to 4th Precinct.

You could configure a system to describe every element I've just described all at once.

Step 1: Host webdata.exe on your web server

The first thing you'll need to do is host webdata.exe on your web server to provide the return URL for your API or WCTP host. The following steps are for IIS but this can be done with any web server suite.

- 1) Copy webdata.exe from the PageGate program directory to your server's scripting directory.
- 2) Open the IIS Admin and select the server object.
- 3) Open Handler Mappings.
- 4) Right click CGI-exe, left click Edit Feature Permissions

Two Way Messaging and Reply Processing

- 5) Check Execute
- 6) Click OK.
- 7) Select the Server object again.
- 8) Open ISAPI and CGI Restrictions.
- 9) Right click in the box and select Add.
- 10) Specify the following:
 - ISAPI or CGI path:** Specify the location of webdata.exe
 - Description:** PageGate API Handler
 - Check 'Allow extension path to execute'
- 11) Click OK.

Step 1a: Share the folder webdata.exe is hosted in

If PageGate is not running on the server hosting webdata.exe, you will need to share the directory where webdata.exe is hosted to allow the [GetASCII interface](#) to parse replies.

Step 2: Configure GetASCII to process the received data

Each API and protocol will have a unique structure to the data returned to your server and you'll need to tell PageGate how to interpret the data and what to do with it in general.

If you own the GetASCII API, you can feed the reply information through that system for advanced processing. This allows you to modify message content and who the replies should go to but does not allow for conditional modification or conditional sending.

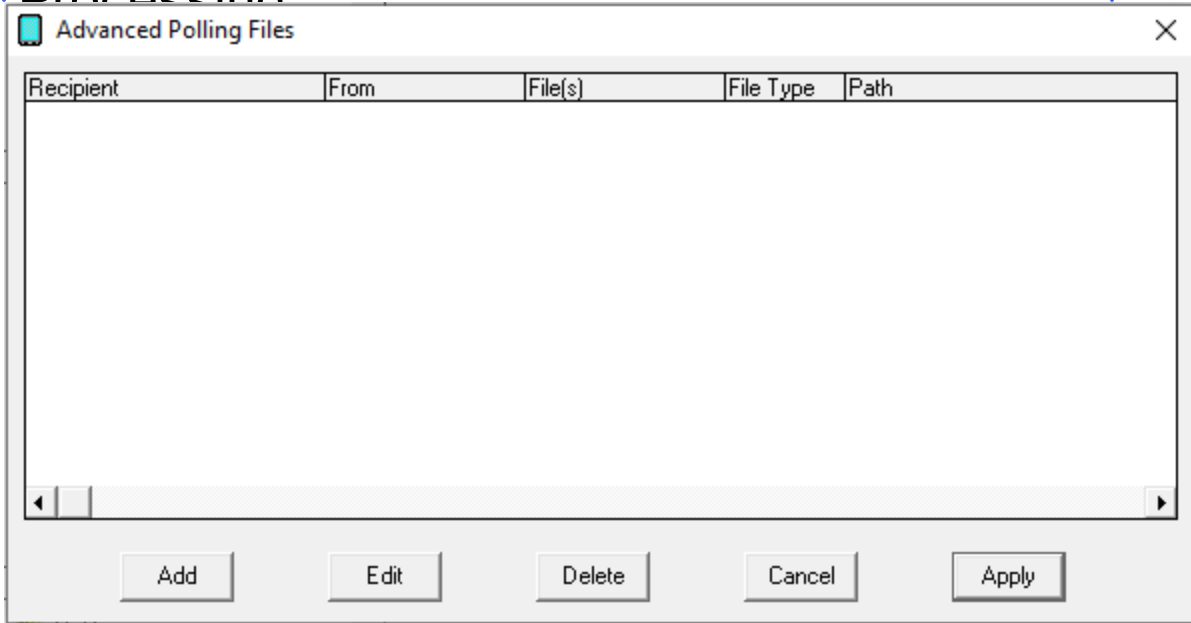
For example, you couldn't have the program scan the body of the message for a keyphrase, then completely change who the message is intended to go to based on it. That would require the [PageGate Filter Pack](#).

To start, you'll want to configure an 'Advanced Polling' rule:

- 1) Open PG Admin.
- 2) Expand Interfaces - GetAscii - Settings - Advanced Polling
- 3) Click Files.

Two Way Messaging and Reply Processing

7



4) Click Add.

5) Specify the following and click Apply:

Recipient: *

From: Leave Blank

File(s): *.asc

Path: The path to the folder webdata.exe is stored in on the web server.

Example: \\WebServer01\pgscripts\

File Type: When using a filter script, set this to Standard.

When not using a filter script, set this to Custom

NOTE: If you will be implementing a [filtering system](#), you will do so at this point and will not need to go through the remaining steps. For more information, please see [PageGate's Filter Script documentation](#).

6) Go to Interfaces - GetAscii - Settings - Advanced Polling - Record

7) Specify the following and click Apply:

Start Pattern: *

Start Offset: 0

End Pattern: ~

End Offset: 0

8) Go to Interfaces - GetAscii - Settings - Advanced Polling - Recipient. This section tells PageGate where in the data to look for the recipient or group this reply should go to.

Two Way Messaging and Reply Processing

9) Select 'Dynamic Recipient' and configure the parameters.

For example, let's say you're working with the WCTP protocol and want to auto-respond to the person who sent the message. In WCTP, you can key off of senderID="Value".

As another example, let's say you're receiving an XML form and one of the tags will determine who needs to receive the reply. You would key off of <XML Tag>Value</XML Tag>.

Start Pattern: This value determines the first character in the recipient block.

For example, if you're working with WCTP and want to key off of senderID="Value". You would use senderID=" as the Start Pattern.

As another example, if you're working with XML and want to key off of <XML Tag>Value</XML Tag>. You would use <XML Tag> as the Start Pattern.

Start Offset: The number of characters in the start pattern.

End Pattern: This value determines the last character in the recipient block.

For example, if you're working with WCTP and are using senderID=" as the Start Pattern, you would use " as the End Pattern.

As another example, if you're working with XML are using <XML Tag> as the Start Pattern, you would use </XML Tag> as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern.

Examples: -1, -10, etc

10) Go to Interfaces - GetAscii - Settings - Advanced Polling - Sender. This section tells PageGate where in the data to look for the sender value.

11) For most implementations, you can select 'Static Sender' and specify a sender value.

However, if the sender's name will be contained in the data read, you can select 'Dynamic Sender' and parameters are configured similarly to the recipient section.

Start Pattern: This value determines the first character in the sender block.

For example, if you're working with WCTP and want to use the value in senderID="Value". You would use senderID=" as the Start Pattern.

As another example, if you're working with XML and want to key off of <XML Tag>Value</XML Tag>. You would use <XML Tag> as the Start Pattern.

Start Offset: The number of characters in the start pattern.

Two Way Messaging and Reply

Processing

End Pattern: This value determines the last character in the sender block.

For example, if you're working with WCTP and are using senderID=" as the Start Pattern, you would use " as the End Pattern.

As another example, if you're working with XML are using <XML Tag> as the Start Pattern, you would use </XML Tag> as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern.

Examples: -1, -10, etc

12) Go to Interfaces - GetAscii - Settings - Advanced Polling - Message. This section tells PageGate what in the data should be considered the message and also allows for basic reformatting of messages.

13) Select 'Dynamic Message' and configure the parameters. If you want everything in the received data to be considered the message, you would use a Start Pattern of * and an End Pattern of ~.

Start Pattern: This value determines the first character in the message block.

For example, if you're working with WCTP and want to use the value in the <wctp-Alphanumeric> tag, you would use <wctp-Alphanumeric> as the Start Pattern.

As another example, if you're working with XML and want to key off of <XML Tag>Value</XML Tag>. You would use <XML Tag> as the Start Pattern.

Start Offset: The number of characters in the start pattern.

End Pattern: This value determines the last character in the sender block.

For example, if you're working with WCTP and are using <wctp-Alphanumeric> as the Start Pattern, you would use </wctp-Alphanumeric> as the End Pattern.

As another example, if you're working with XML are using <XML Tag> as the Start Pattern, you would use </XML Tag> as the End Pattern.

End Offset: The negative value of the number of characters in the End Pattern.

Examples: -1, -10, etc

The [lookup table](#) enables PageGate to perform basic search and replace commands. This allows you to interchange or translate values as needed.

To enable this feature, check Use Lookup Table and click Edit Table.

For more information, please see the [lookup table](#) section of the documentation.

Reliability in messaging is incredibly important, so PageGate has two different styles of failover and a synchronization system that can be configured.

[Carrier level failover](#)

This Carrier level failover allows PageGate to automatically and seamlessly transition between communication methods in the event that it has trouble delivering a message.

As an example, let's say you're using a cellular modem to deliver your messages. You would probably also want to configure an internet based carrier that the cellular modem could fail over to in the event of hardware failure.

So, let's say that you're using a cellular modem and that you're also registered for Verizon's Enterprise Messaging Access Gateway (EMAG) WCTP API. You could have the cellular modem fail over to Verizon's EMAG WCTP host, so that in the event of hardware failure, your traffic still has a viable route of delivery available.

[Click here](#) for steps to configure a Carrier level failover.

[Recipient level failover](#)

The Recipient level failover allows PageGate to automatically deliver a message to an alternate address, location or person in the event that the primary contact couldn't be messaged.

For example, let's say that we have John Smith's Cel and John Smith's Email as recipients in PageGate. We could have John Smith's Cel fail over to John Smith's Email or we could configure the reverse. By doing so, we're telling PageGate that if a message can't be delivered to John Smith's cell phone, that message should be delivered to his email instead or vice versa.

As another example, let's say we have a recipient called Failure Notification that's tied to a network administrator's email address. If we specify Failure Notification as the failover for every recipient configured, if any message fails, that message will instead be delivered to the network administrator in our example. With PageGate's templating system, we could also prefix the text delivered to the administrator to say "The following message could not be delivered for %OriginalRecipient%".

[Click here](#) for steps to configure a Recipient level failover.

Just to note, PageGate doesn't limit the number of failovers you can configure and will allow you to fail over as many times as your environment and infrastructure support.

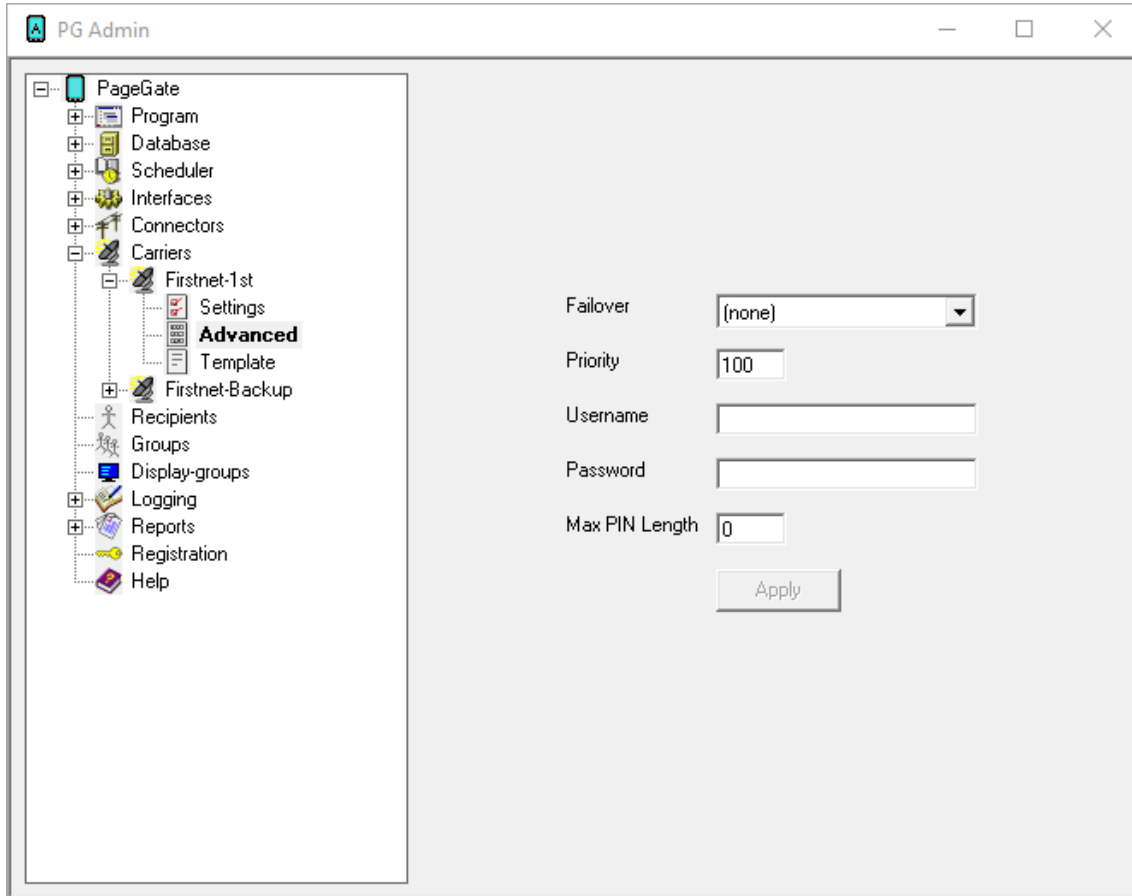
To configure multiple PageGate servers to synchronize their data, [please reference the Synchronization](#) section of the documentation.

Carrier Level Failover

- 1) Open the PageGate Admin.
- 2) Configure your primary and failover carriers.



- 3) Open the Advanced sub-section of your primary carrier.



Failover Scenarios

4) In the Failover field, select your failover carrier.

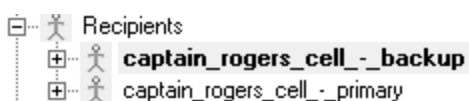
A configuration dialog box with the following fields:

- Failover: Firstnet-Backup (dropdown menu)
- Priority: 100 (text input)
- Username: (empty text input)
- Password: (empty text input)
- Max PIN Length: 0 (text input)
- Apply (button)

5) Click Apply.

Recipient Level Failover

- 1) Open the PageGate Admin.
- 2) Configure your primary and failover recipients.



3) Go in to the Settings section of your primary recipient.

The PG Admin interface shows the following configuration for the primary recipient:

- Full Name: Captain Rogers Cell - Primary
- Recipient: captain_rogers_cell_-_primary
- Enabled Services:
 - GetAscii
 - GetMail
 - GetSerial
 - GetTap
 - GetWeb
 - GUI Client
- Type: Normal (dropdown)
- Carrier: Firstnet-1st (dropdown)
- Failover: (none) (dropdown)
- ID/PIN: 123 (text input)
- Max Chars: 999 (text input)
- Apply (button)

4) In the Failover field, select your failover recipient.

Full Name: Captain Rogers Cell - Primary
Recipient: captain_rogers_cell_-_primary

Enabled Services

GetAscii GetMail GetSerial
 GetTap GetWeb GUI Client

Type: Normal
Carrier: Firstnet-1st
Failover: captain_rogers_cell_-_backup
ID/PIN: 7818290500
Max Chars: 999

Apply

5) Click Apply.

The PageGate GUI Client is a Windows based program that interacts with PageGate's database.

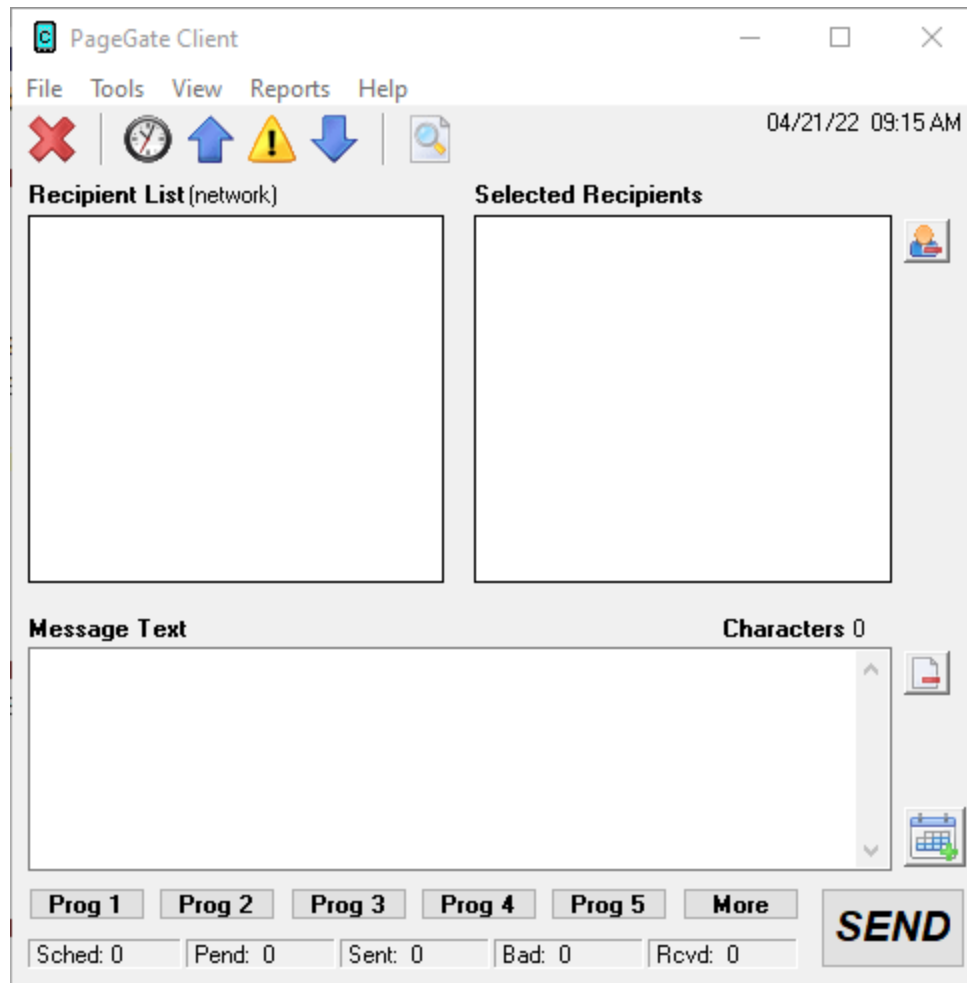
When using an MSJet database (default), the PageGate Client will need access to the PageGate server's database directory. You will need to share that folder on the server and provide all appropriate users and usergroups the read, write and modify permissions.

When using an SQL database, the PageGate Client will need a configuration file that tells it how to interact with the SQL server.

[Recipients](#) and [groups](#) configured in [PageGate Admin](#) display in the recipient list, which allows a user to select them to send a message immediately or schedule a message to go out at a later time.

If [two-way messaging](#) has been configured, GUI Clients also have the ability receive replies to messages sent, reply to any received messages and message other workstations in a chat style interaction.

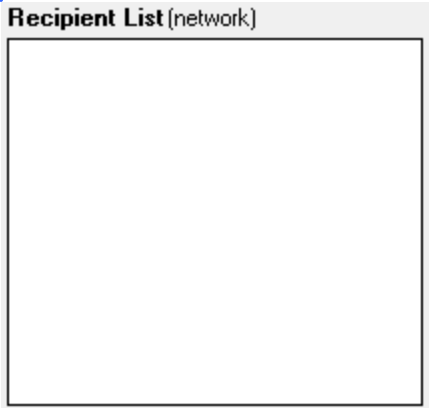
It is also possible to have a GUI Client [reference multiple PageGate servers](#).



This API offers a few unique features, such as the ability to create a scheduled message to go out and to create a scheduled message to go out at a repeating interval. The PageGate server application must be installed and configured before the PageGate GUI Client can be used. Please refer to the PageGate Server's help file for more information on the installation, configuration, and operation of the PageGate Server.

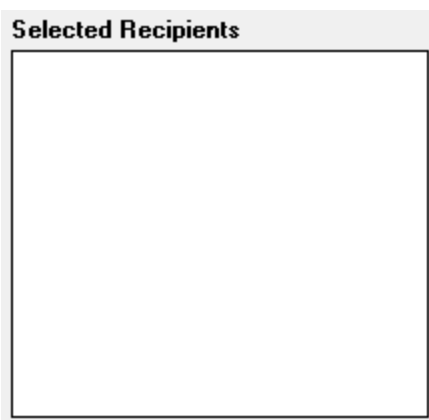
The first step in using the GUI Client is understanding its interface. There sections of the main interface are:

- **Recipient List**



This section displays all recipients and groups available to message from this installation of the GUI Client. This list can be controlled by [display groups](#), which allow administrators to limit which recipients and groups are available for this workstation to message, as well as modified by [filtering the list](#) or creating a [local recipient list](#).

- **Selected Recipients**



This section displays all recipients and groups selected. To select a recipient or group to message, left click on their name in the 'Recipient List'.







- **Message Text**



This section is where you'll type the message you want to deliver.

- **Buttons**

Along the top of the interface, you'll see these buttons:

	This button cancels all pending messages.
	This button displays a list of messages currently in the pending queue for immediate delivery.
	This button displays a list of messages successfully sent by this workstation's station ID .
	This button displays a list of messages that were not successfully sent by this workstation's station ID .
	This button displays the two-way messaging interface.
	This button displays the history of messages (good and bad) sent by this workstation's station ID .

Along the bottom of the interface, you'll see these buttons:



These buttons allow you to configure [preset messages](#), which configure pre-determined messages to potentially save your operators a lot of typing.

Along the right hand side, you'll see these buttons:



- **Clear Selected Recipients**

Clicking this button next to the 'Selected Recipients' section will clear all selections from the list.



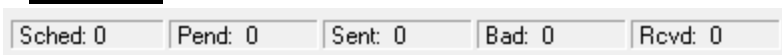
- **Clear Message**

Clicking on this button next to the 'Message Text' section will clear anything typed in that section.

- **Options**

This button is used to configure [scheduled and repeating messages](#).

- **Statuses**



This section displays the status of all traffic a workstation is allowed to see and the tickers update in real time. The categories listed are:

- Sched

This is the count of messages scheduled from this workstation to go out at a later date and time.

- Pend

This is the count of messages in the pending queue for immediately delivery.

- Sent

This is the count of messages successfully delivered by this workstation.

- Bad

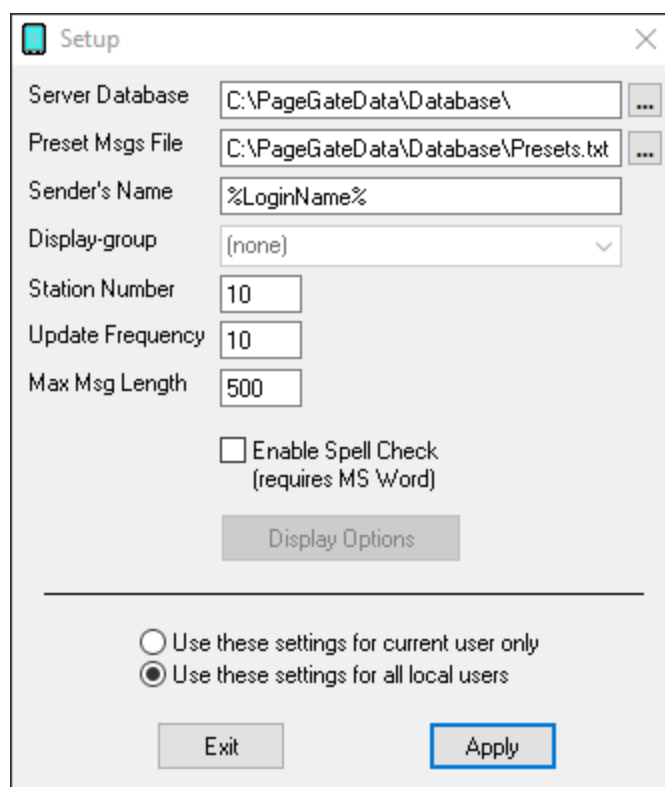
This is the count of messages that were not successfully delivered.

- Rcvd

If this workstation is configured to receive replies, this will display the number of reply messages received.

MSJet Database Setup (default)

After [installing the PageGate Client](#), the very first time you run the program, you will be prompted for some basic information.



Setting	Function
Server Database	This should be the path to the PageGate Server's Database. This can be either a local path (if you're running the GUI Client on the same system as the PageGate Server) or a network path.

Preset Msgs File	The PageGate Client's preset messages are stored in the text file specified in this path. Workstations that point to the same file will share presets. Workstations that point to their own file will have their own presets.
Sender's Name	<p>This field determines the sender's name attached to any messages sent from this GUI Client. By default, this field is set to %LoginName%, which is a value that uses the Windows username of whoever is currently logged in to the operating system.</p> <p>This field can also be set to a static value such as Workstation1, Sales, Hestia, Lamar, etc.</p> <p>If you want the ability to receive reply messages to this workstation, this field MUST match the name of a local recipient that is referenced by the replies system.</p>
Display-group	<p>This option will lock this installation of the PageGate Client to the selected display group. This will only allow the client to display recipients and groups an administrator has added to the display group.</p> <p>To use this option, you must first create and populate a display group in the PageGate Admin.</p>
Station Number	<p>This field provides an identifying number that all traffic sent from this system will have associated with it.</p> <p>Workstations that share a station number will be able to see and report on each others' messages. Workstations that do not share a station number will only be able to report on their own messages.</p>
Update Frequency	This field determines how often (in seconds) the GUI Client posts data to the PageGate Database.
Max Msg Length	This field determines how many characters can be typed in the Message field of the GUI Client.
Enable Spell Check	By enabling this option, the GUI Client will automatically spellcheck any messages typed. This option requires MS Word to be installed.
Display Options	This option isn't something you can configure initially. After applying the initial configuration, though, clicking this button will open the Display Options for this GUI client workstation.
Use these settings for current user only	Set this option if each Windows User on this workstation needs to have their own copy of the GUI Client configured.
Use these settings	Set this option if you all Windows Users who log in to this workstation should share a common configuration.

for all local users	
------------------------	--

SQL Database Setup

When using an SQL database hosted on a cloud server, if you enable 'Always Encrypted' in SQL, the PageGate Admin and GUI Client applications can be installed on any computer and given access to your PageGate server. This makes connecting remote workstations far easier as they can be given direct access.

After [installing the PageGate Client](#), you will need to place a configuration file in the PageGate Client's database directory that tells it how to connect to the SQL server.

The file must be named `SqlServerConnectionString.txt` and will be formatted based on the style of SQL authentication being used.

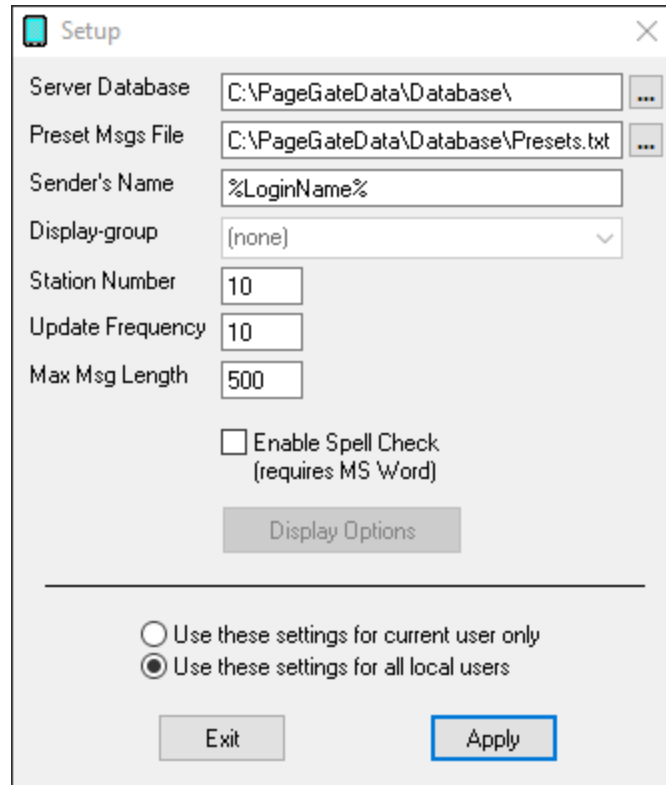
When using Domain user/windows user authentication, use this:

```
ODBC;  
Driver=SQL Server;  
Server=tcp:PUT-SQL-SERVER-IP-ADDRESS-HERE,1433;  
Database=The name of the SQL database;
```

When using SQL user authentication, use this:

```
ODBC;  
Driver=SQL Server;  
Server=tcp:PUT-SQL-SERVER-IP-ADDRESS-HERE,1433;  
Database=The name of the SQL database;  
uid=SQLUsername;  
pw=SQLPassword;
```

After placing the configuration file in the folder, run the PageGate Client and you will be prompted with the first time setup screen.



Setting	Function
Server Database	This should be the path to the SqlServerConnectionString.txt file. This may reside on a network or local path.
Preset Msgs File	The PageGate Client's preset messages are stored in the text file specified in this path. Workstations that point to the same file will share presets. Workstations that point to their own file will have their own presets.
Sender's Name	<p>This field determines the sender's name attached to any messages sent from this GUI Client. By default, this field is set to %LoginName%, which is a value that uses the Windows username of whoever is currently logged in to the operating system.</p> <p>This field can also be set to a static value such as Workstation1, Sales, Hestia, Lamar, etc.</p> <p>If you want the ability to receive reply messages to this workstation, this field MUST match the name of a local recipient that is referenced by the replies system. For more information, please see the PageGate Server documentation.</p>
Display-group	This option will lock this installation of the PageGate Client to the selected display group. This will only allow the client to display recipients and groups an administrator has added to the display group.

	To use this option, you must first create and populate a display group in the PageGate Admin.
Station Number	This field provides an identifying number that all traffic sent from this system will have associated with it. Workstations that share a station number will be able to see and report on each others' messages. Workstations that do not share a station number will only be able to report on their own messages.
Update Frequency	This field determines how often (in seconds) the GUI Client posts data to the PageGate Database.
Max Msg Length	This field determines how many characters can be typed in the Message field of the GUI Client.
Enable Spell Check	By enabling this option, the GUI Client will automatically spellcheck any messages typed. This option requires MS Word to be installed.
Display Options	This option isn't something you can configure initially. After applying the initial configuration, though, clicking this button will open the Display Options for this GUI client workstation.
Use these settings for current user only	Set this option if each Windows User on this workstation needs to have their own copy of the GUI Client configured.
Use these settings for all local users	Set this option if you all Windows Users who log in to this workstation should share a common configuration.

Using the GUI Client

The PageGate GUI Client is used for scheduling and sending messages to the PageGate Server's list of Recipients and Groups. If [two-way messaging](#) has been configured, GUI Clients also have the ability receive replies to messages sent, reply to any received messages and message other workstations in a chat style interaction.

The PageGate Client can be run from the PageGate server or on any workstation that has network access to the PageGate server's Database and displays a list of available recipients and groups and is highly customizable.

Recipients can be moved to the Selected Recipients column simply by clicking on their name. The Message Text field is where you type the message you want to send.

Along the bottom, you'll notice the Prog 1 through Prog 5 and More buttons, those are [preset messages](#) and can be configured many ways. For information on what the preset

messages are and how to configure them, please see the [preset messages](#) section of the documentation.

- [File](#)

This section provides access to the PageGate Client's configuration and settings.

- [Tools](#)

This section provides methods of filtering the recipient list and provides a spell-check function.

- [View](#)

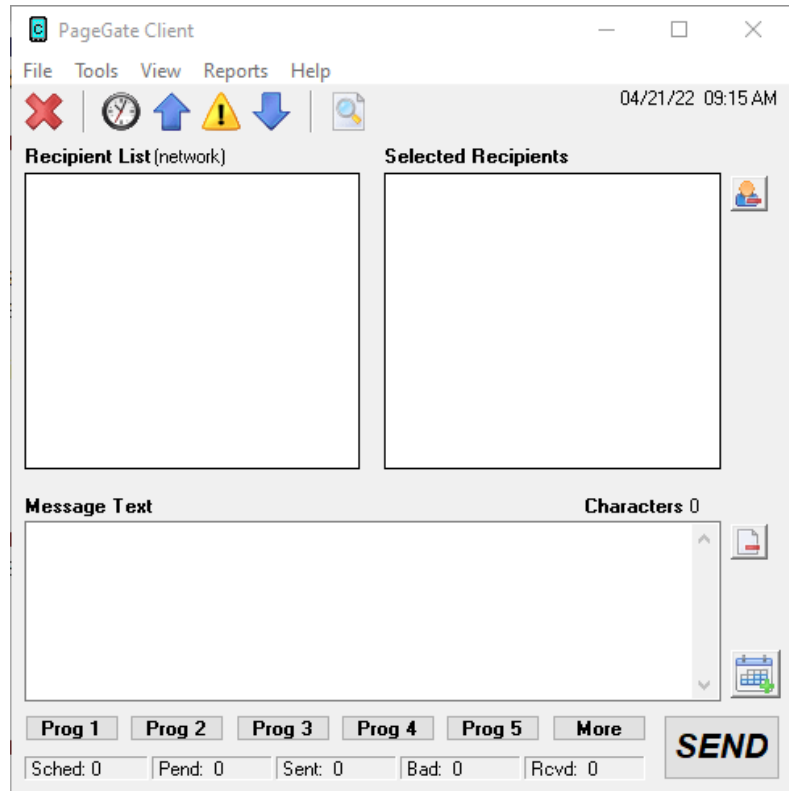
This section provides the ability to view and review traffic in the pending queue or sent from this workstation.

- [Reports](#)

This section allows you to run reports on messages sent from any GUI Client that shares a [station number](#) with this Client.

- [Help](#)

This section provides basic help and assistance for using the PageGate Client.



Scheduling Messages

This section allows you to configure scheduled and repeating messages. Before clicking on the 'Options' button, select a recipient (or group) and type the message you want scheduled in the 'Message Text' box. Before you click the Send button, click the Options button to configure your schedule. When you've finished, click Apply, then click the Send button.

The screenshot shows a dialog box titled "Advanced Message Options". It contains three main sections for scheduling a message:

- Send Message When:** Date is 04 / 21 / 2022, Time is 01 : 01 PM.
- Resend Message Every:** Days, Hours, and Minutes are all empty input fields.
- Stop Sending Message:** Date is 04 / 21 / 2022, Time is 01 : 01 PM.

At the bottom of the dialog are two buttons: "Close" and "Apply".

With this prompt, you can schedule messages to automatically be sent at some point in the future. You can also determine if that message will repeat or not.

To schedule a page to be sent at a future date and time, fill in the 'Send Message When' fields.

To have the scheduled message repeat, set the "Resend Messages Every" fields to the interval at which the message should be delivered.

Last, fill in the 'Stop Sending Message' fields to set the date and time for the message to stop being sent.

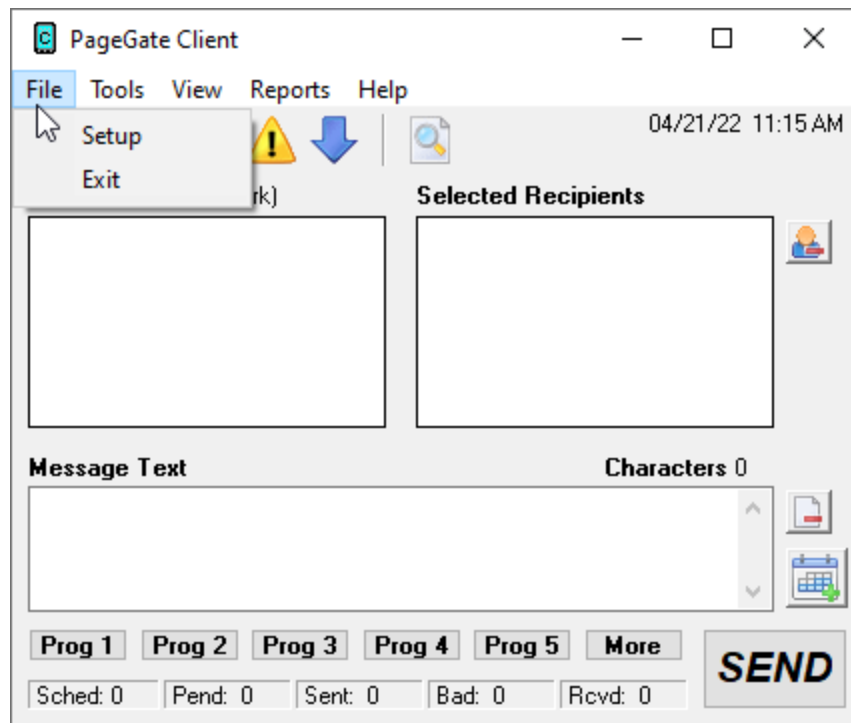
For example, let's say that you had a message that, starting tomorrow, needed to be sent out once a day for one month.

You would set the "Send Message When" date to tomorrow's date, set the time for whatever time the message needs to be delivered. You would then set the "Resend message every" to 1 day OR 24 hours OR 1440 minutes.

The final step would be to set the "Stop Sending Message" date and time to exactly one month from the "Send Message When" Date.

Click Apply, then click on the Send button. The message is now scheduled!

File



- [Setup](#)

This option allows you to configure the PageGate Client's settings, including specifying a display group to limit what recipients and groups this workstation is allowed to display.

- [Exit](#)

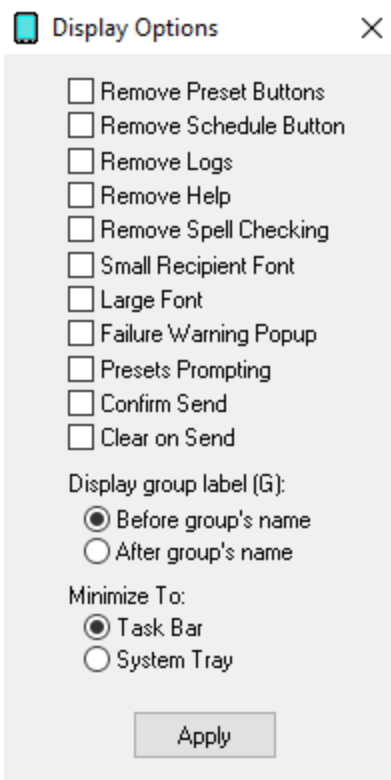
This option closes the PageGate Client.

Setup

Setting	Function
Server Database	This should be the path to the PageGate Server's Database. Typically speaking, this will be a mapped drive or UNC path to the PageGate server's network shared database directory.
Preset Msgs File	The PageGate Client's preset messages are stored in the text file specified in this path. Workstations that point to the same file will share presets. Workstations that point to their own file will have their own presets.
Sender's Name	This field determines the sender's name attached to any messages sent from this GUI Client. By default, this field is set to %LoginName%, which is a value that uses the Windows username of whoever is currently logged in to the operating system. This field can also be set to a static value such as Tech, Sales, Lamar, etc.
Display-group	This option will lock this installation of the PageGate Client to the selected display group. This will only allow the client to display recipients and groups an administrator has added to the display group. To use this option, you must first create and populate a display group in the PageGate Admin .
Station Number	This field provides an identifying number that all traffic sent from this system will have associated with it. Workstations that share a station number will be able to see and report on each others' messages. Workstations that do not share a station number will only be able to report on their own messages.

Update Frequency	This field determines how often (in seconds) the GUI Client posts data to the PageGate Database.
Max Msg Length	This field determines how many characters can be typed in the Message field of the GUI Client.
Enable Spell Check	By enabling this option, the GUI Client will automatically spell check any messages typed. This option requires MS Word to be installed.
Display Options	This button allows you to modify basic display settings for the PageGate Client.
Use these settings for current user only	Set this option if each Windows User needs to have their own copy of the GUI Client configured.
Use these settings for all local users	Set this option if you want these settings to be saved for all Windows Users who log in to this workstation.

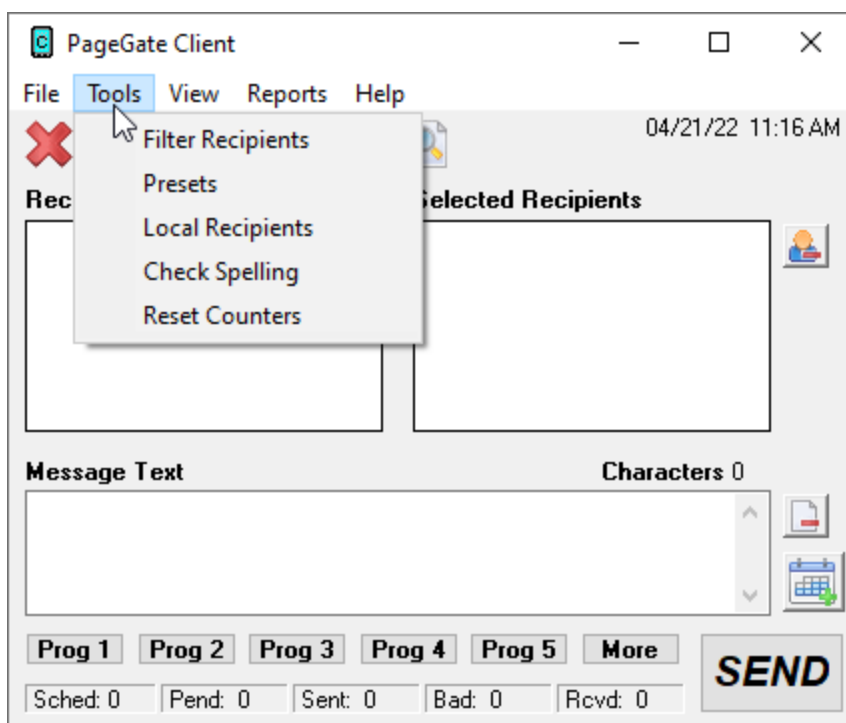
Display Options



Setting	Function
---------	----------

Remove Preset Buttons	Enabling this will remove the Prog1, Prog2, Prog3, Prog4, Prog5, and More buttons. This removes this workstation's ability to use preset messages .
Remove Options Button	Enabling this will remove the Options button. This removes this workstation's ability to schedule messages .
Remove Logs	Enabling this will remove the ability of this GUI Client to view any log files. This removes this workstation's ability to run reports .
Remove Help	Enabling this will remove the Help menu.
Remove Spell Checking	Enabling this will remove the Spell Checking function of this GUI Client.
Small Recipient Font	Enabling this will display the recipient and selected recipient lists in a smaller font.
Large Font	Enabling this will display the recipient and selected recipient lists in a larger font.
Failure Warning Popup	When this is enabled, every time a message fails, you will be notified by a popup window.
Presets Prompting	<p>For this setting to function properly, you must first configure a set of preset messages in the PageFate client. Next, set up one of the first nine presets with a recipient but no message.</p> <p>If that has been done and this option is enabled, if you click on the visible preset button, a pop-up will list all of the possible preset messages for that recipient. This allows a user to click a preset for a recipient and choose a commonly used message for them.</p>
Confirm Send	When this is enabled, every time the Send button is clicked, you will be prompted to confirm the delivery of the message.
Clear on Send	When this option is enabled, the Message and Selected Recipient portions of this GUI Client are cleared after the Send button is clicked.
Display group label (G)	You can choose to have the (G) for Group label appear before or after your groups' names. If the (G) appears before, the groups will automatically be sorted to the top of the Recipient list. If the (G) appears after, the groups will be sorted by the first letter in the group name.
Minimize To	You can choose to have the GUI Client minimize to the Task Bar or the System Tray. If the GUI Client is set to minimize to the Task Bar, you will see it to the right of the Start Menu when the program is minimized. When the program is set to minimize to the System Tray, it will appear as an icon of a pager next to the clock in the lower right hand corner of the screen.

Tools



- [Filter Recipients](#)

This section allows you to filter what recipients are currently displayed in the recipient list. This is a useful way for quickly finding recipients and groups.

- [Presets](#)

This section allows you to modify the preset messages.

- [Local Recipients](#)

This option allows you to configure a list of recipients that you want this workstation to default to.

- [Check Spelling](#)

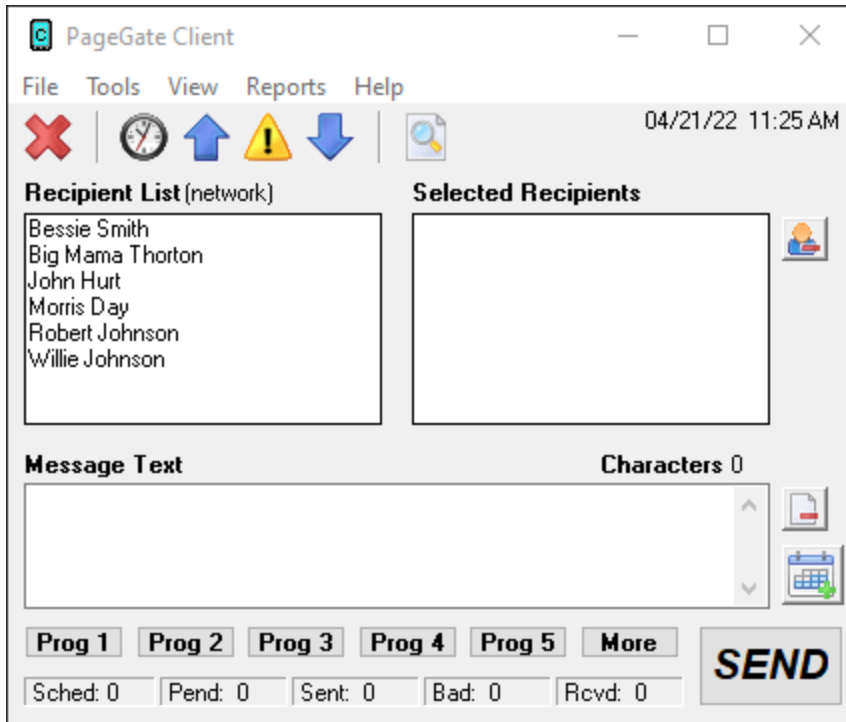
Selecting this option will check spelling and grammar in the Message Text section.

Filter Recipients

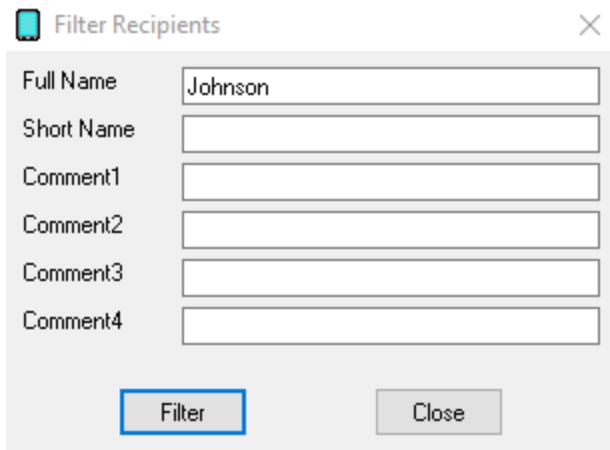
This function allows you to filter the list of recipients shown in the 'Recipient List' of the GUI interface. You can filter recipients by full name (the name that is displayed in the 'Recipient List'), by short name (the name used by the PageGate Admin), or by any of for other user definable fields (these are also configured within the PageGate Admin).

Setting	Function
Full Name	This field requires the Full Name field of a recipient in the PageGate Admin to match the pattern you enter here. For example, to have only those recipients whose Full Name field begins with Tech show in the recipient list, you would enter Tech here.
Short Name	This field requires the Recipient field of a recipient in the PageGate Admin to match the pattern you enter here. For example, to have only those recipients whose Recipient field begins with oncall show in the recipient list, you would enter oncall here.
Comment1 - Comment4	When creating a recipient in the PageGate Admin, you can specify certain comment fields. This field requires the Comment Field in the Recipient's Settings to match whatever you enter here.
Filter	Clicking on this button will Filter the Recipient list with your criteria.
Close	This will close this option dialog without filtering your Recipient view.

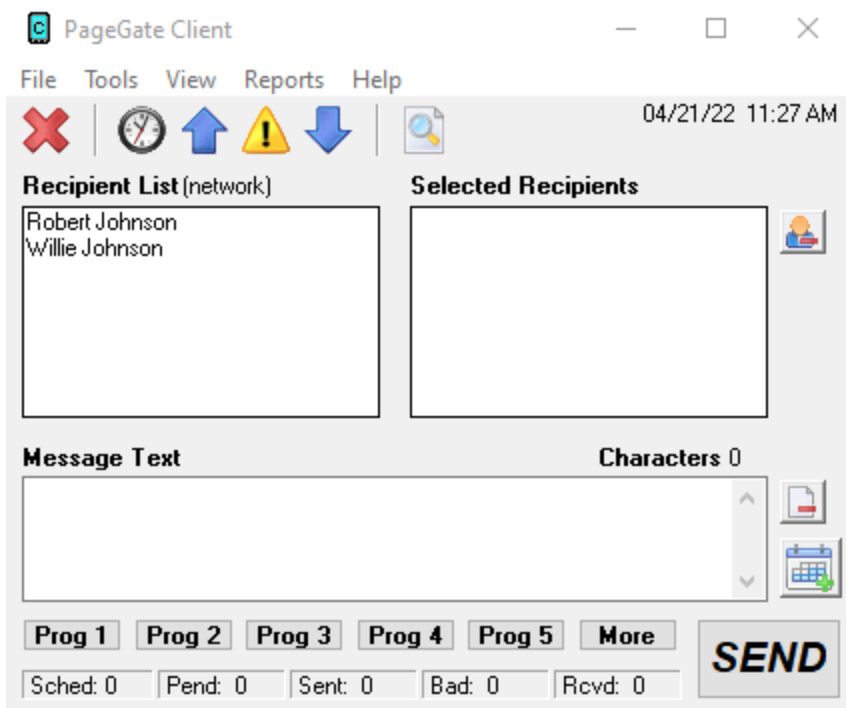
As an example, let's say that this is our recipient list:



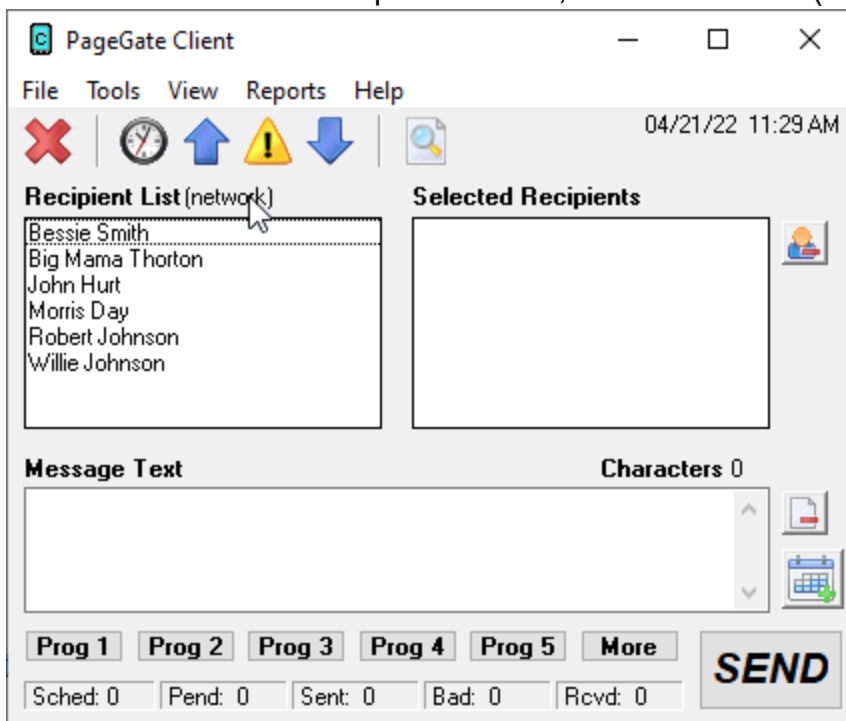
If we set up a filter like this:



That will make the recipient list display only those recipients and groups with 'Johnson' in the Full Name field of their record in the PageGate server's database.

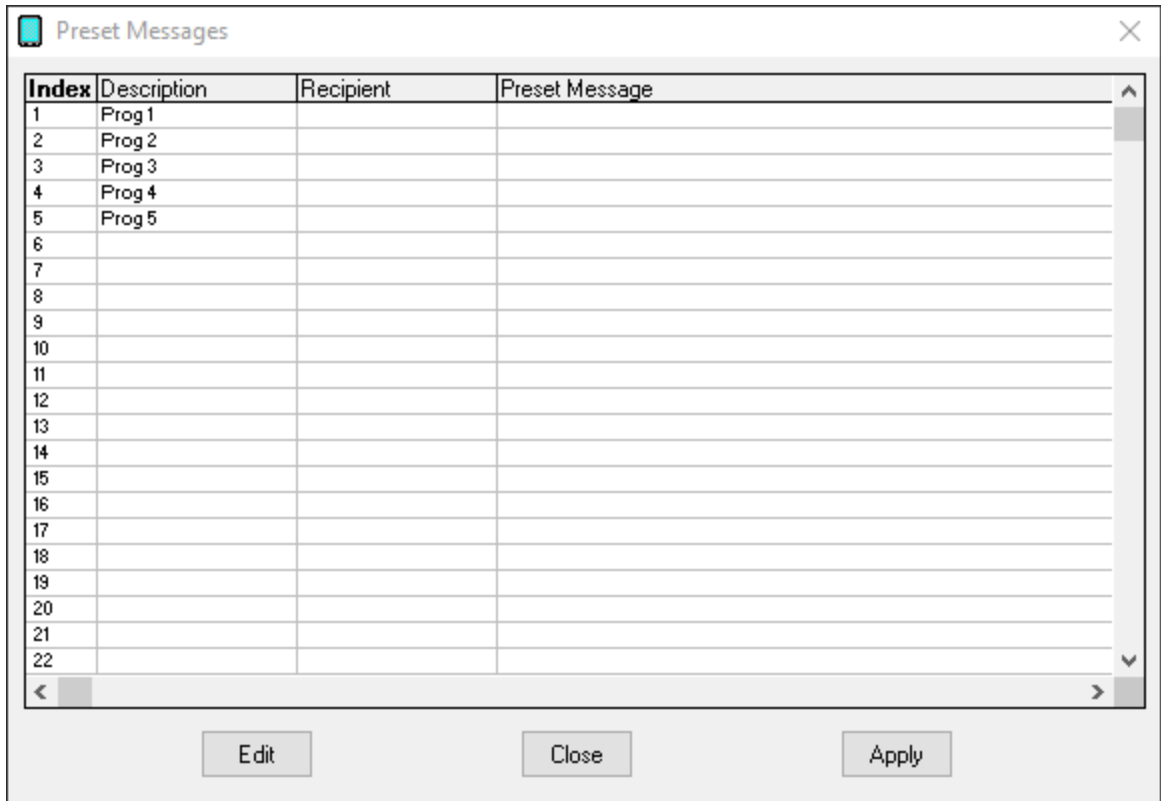


To return to the normal recipient list view, click on the word (network) next to Recipient List.

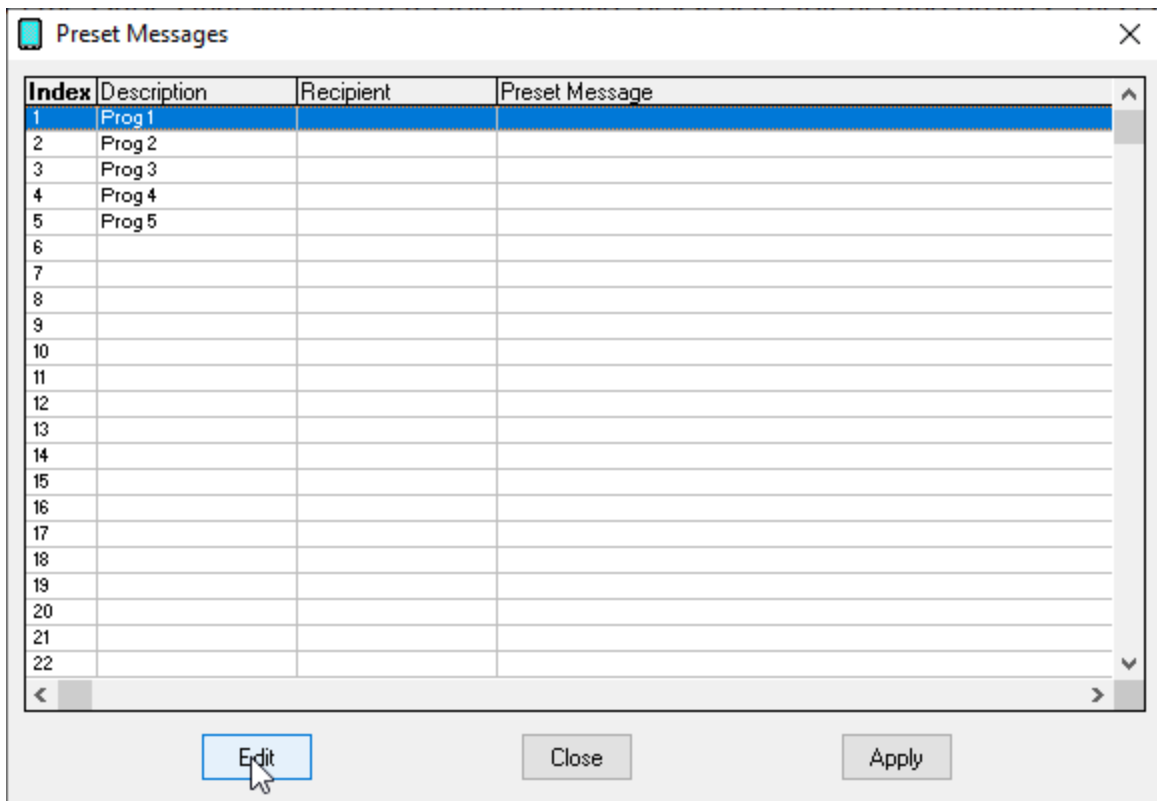


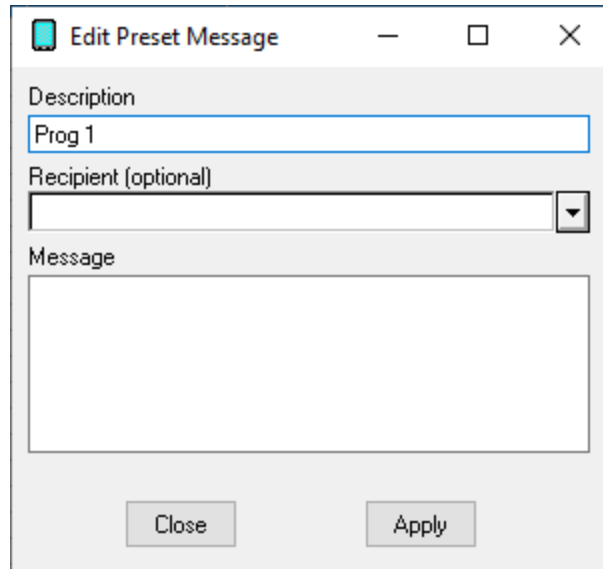
Presets

The Presets function allows you to customize and create preset messages that will go to a recipient, group, or list of recipients and groups. This is useful if you need 'canned' messages and don't want to have to type the same message over and over and over.



To modify a preset message, highlight an entry and click Edit.



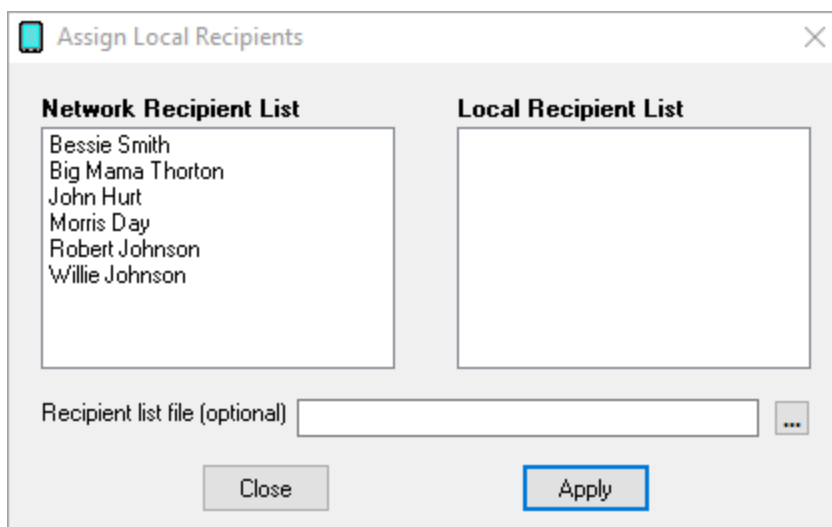


Setting	Function
Description	This field represents a description of this preset message. This field also represents the text displayed in the GUI Client for the first five preset message buttons (Prog 1 through Prog 5, by default).
Recipient (optional)	This field allows you to specify a specific recipient or group this preset message should go to. This saves time by pre-selecting the necessary recipient or group.
Message	This field allows you to specify the body of the message that should be inserted when this preset message is referenced.
Close	This button will close the 'Edit Preset Message' dialog without saving any recently made changes.
Apply	This button will close the 'Edit Preset Message' dialog and will save any recently made changes.

Local Recipients

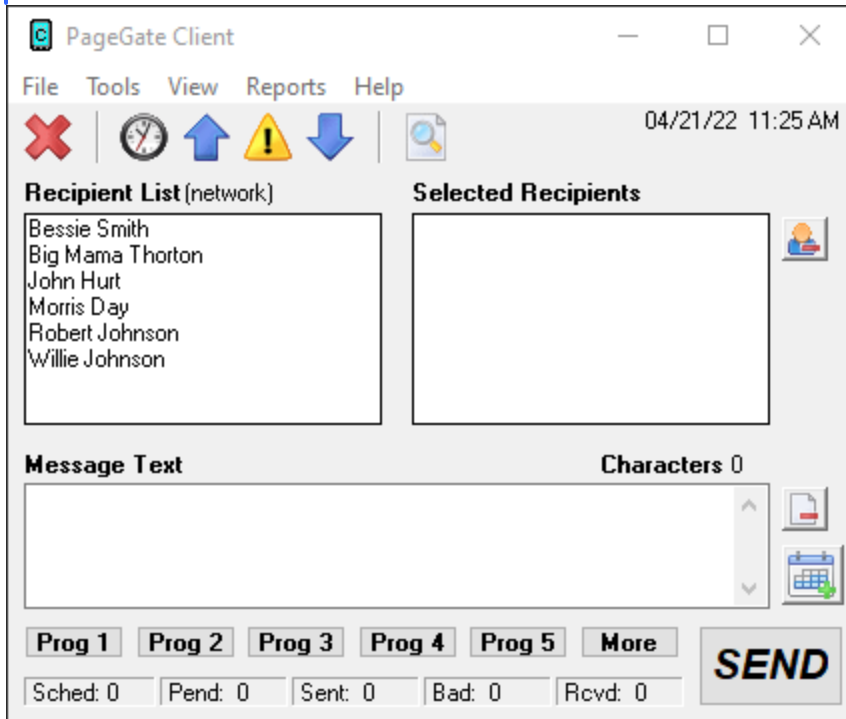
NOTE: If you have a local recipient list configured, it will be the default list displayed each time you start the GUI Client.

This section allows you a user to customize the list of recipients and groups displayed in the Recipients list, allowing a user to easily switch back and forth between their custom defined list and the master list of recipients and groups they're allowed to see.

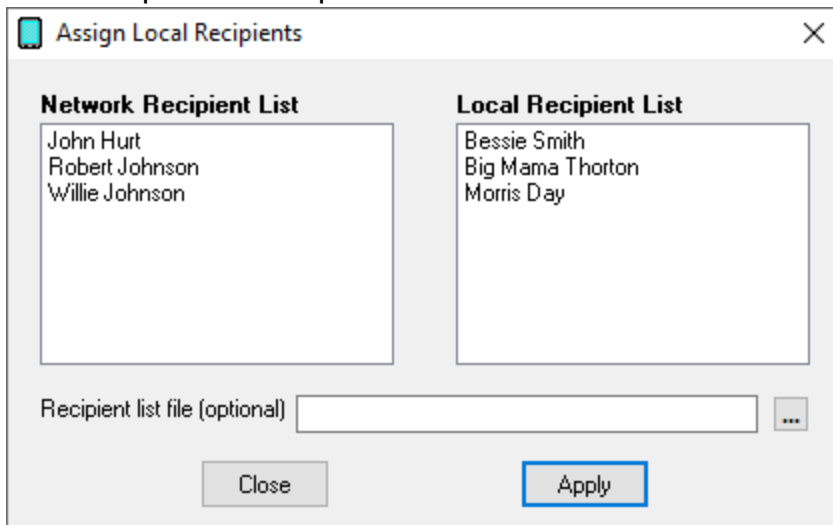


Section	Function
Network Recipient List	This section displays all recipients and groups available to this GUI Client. To move a recipient or group to the 'Local Recipient List' section, click on their name.
Local Recipient List	This section displays the recipients and groups selected for the local recipient list. To move a recipient or group to the 'Network Recipient List' section, click on their name.
Recipient list file	<p>By default, each GUI Client has its own, unique local recipient list. If workstations should share a local recipient list, specify the file path and file name in this field.</p> <p>For example, let's say that we have a shared network location of <code>\PageGateServer\Database\</code></p> <p>You could then specify <code>\\PageGateServer\Database\localrecipients.txt</code> in this field for all GUI Client you want to share a local recipient list.</p>

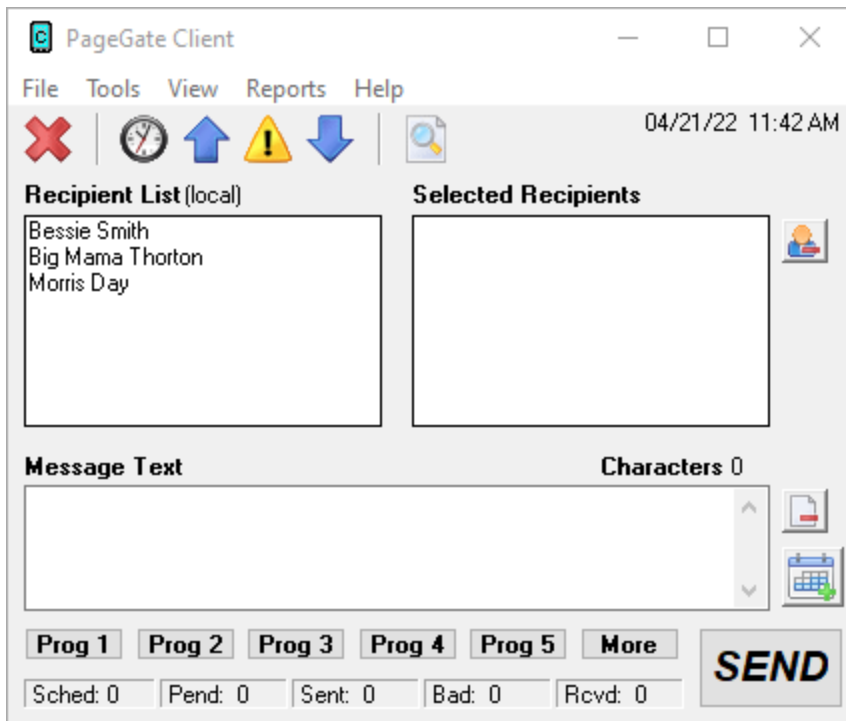
As an example, let's say that this is our recipient list:



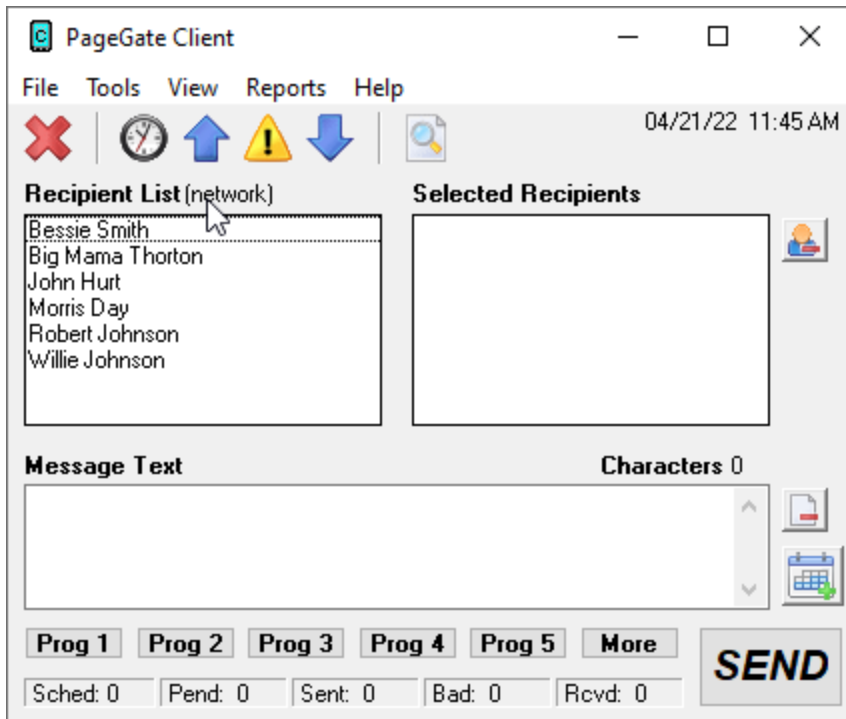
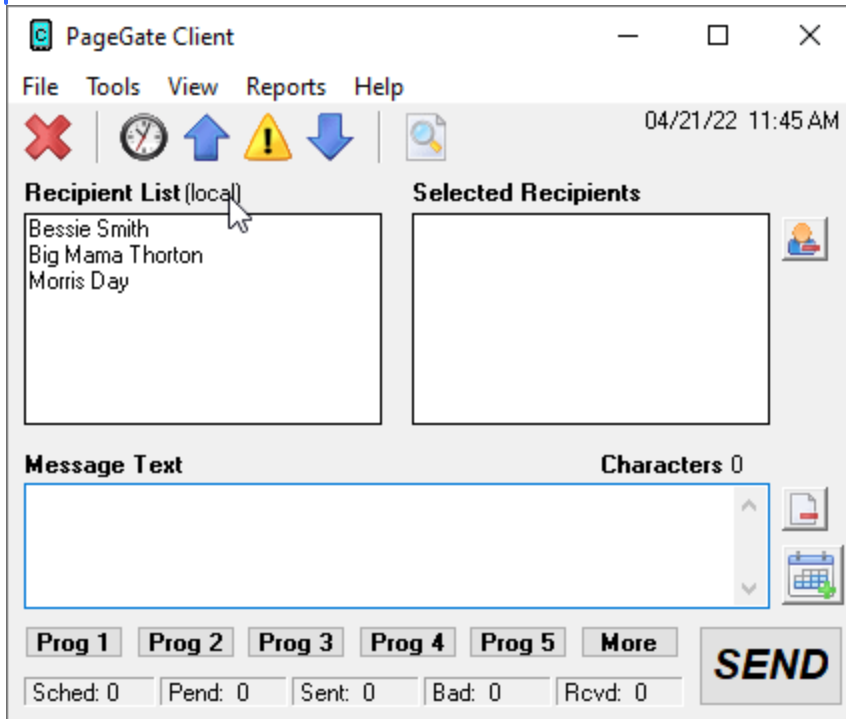
If we set up a local recipients list like this:



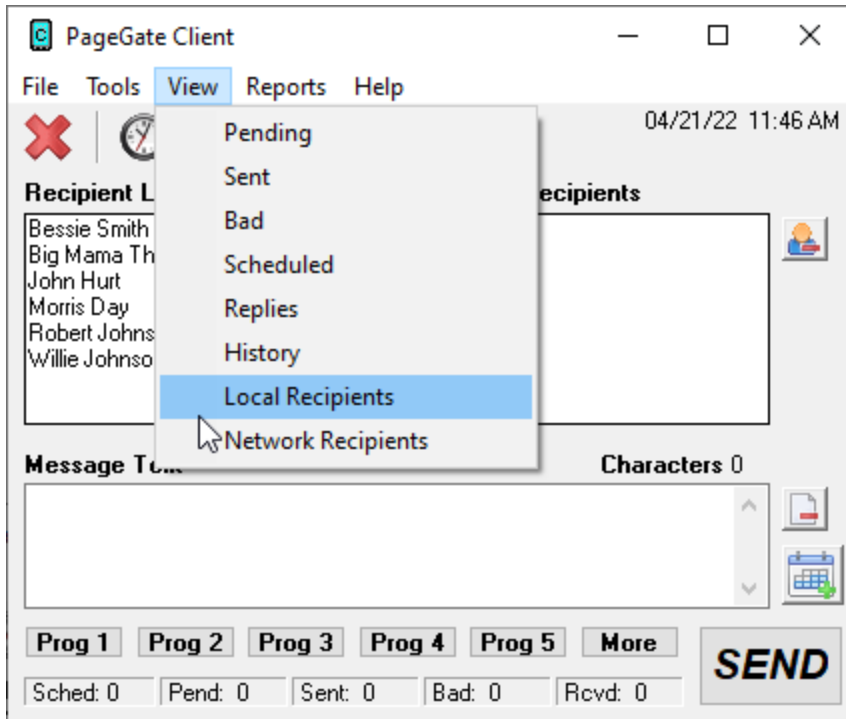
That will make the recipient list display as follows:



To return to the normal recipient list view, click on the word (local) next to Recipient List.

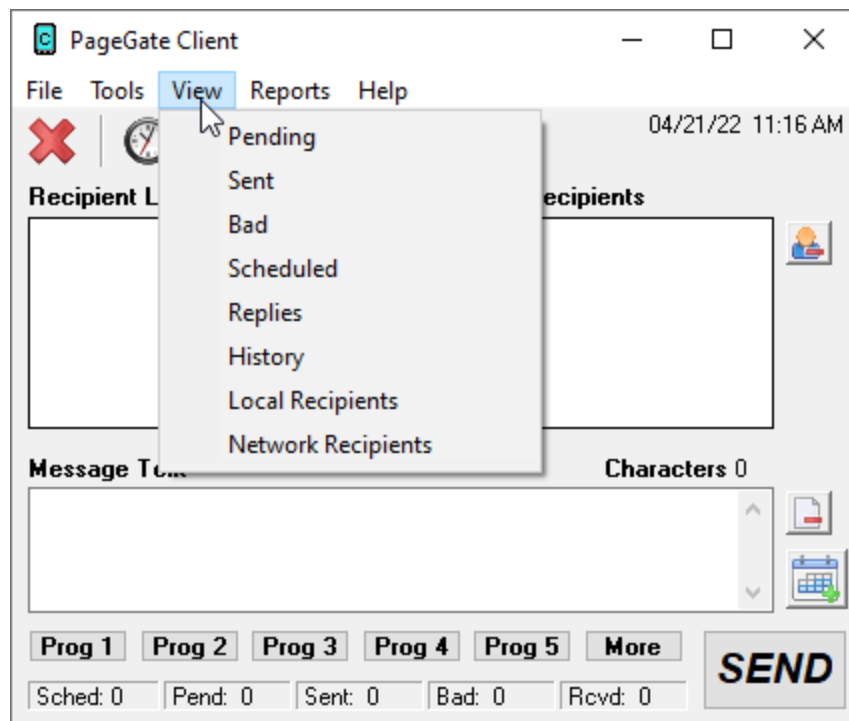


There are also options in the [View menu](#) to switch between the local and network recipients lists.



View

This section provides the ability to review messages sent from this workstation and also provides another option for switching between a [local recipients list](#) and the network recipients list.



- [Pending](#)

This option will display all messages sent from this Client's [station number](#) that are in the pending message queue.

- [Sent](#)

This option will display all messages sent from this Client's [station number](#) that were successfully delivered.

- [Bad](#)

This option will display all messages sent from this Client's [station number](#) that could not be delivered.

- [Scheduled](#)

This option will display all scheduled messages created by this Client's [station number](#).

- [Replies](#)

If [two-way messaging](#) has been configured, this option will display the replies dialog and allow this workstation to send replies to received messages and message other workstations in a chat style interaction.

- [History](#)

This option will display the entire history of messages, good or bad, for this Client's [station number](#).

- [Local Recipients](#)

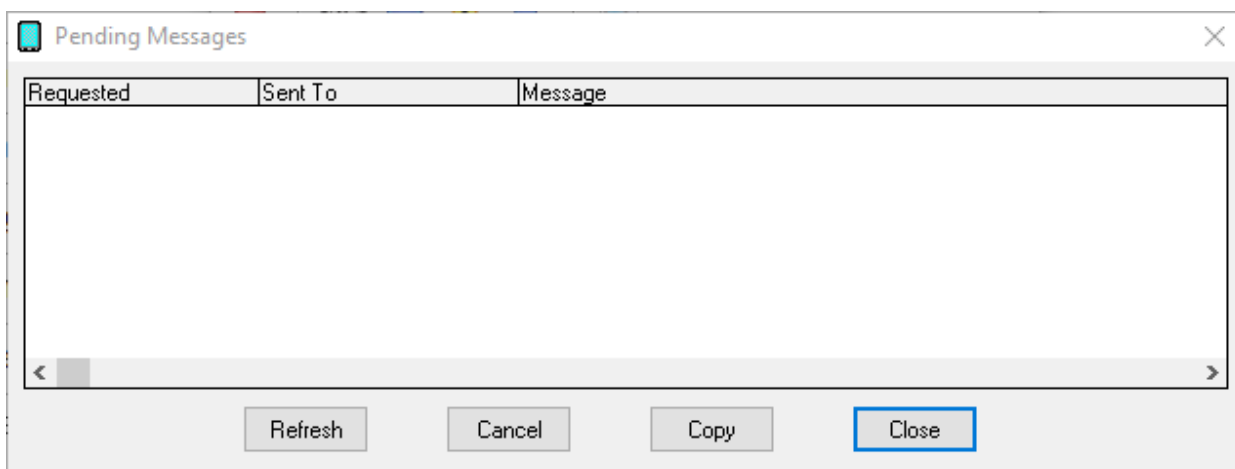
Selecting this option will display any configured [local recipient list](#).

- [Network Recipients](#)

Selecting this option will display the full list of recipients and groups available to this Client.

Pending

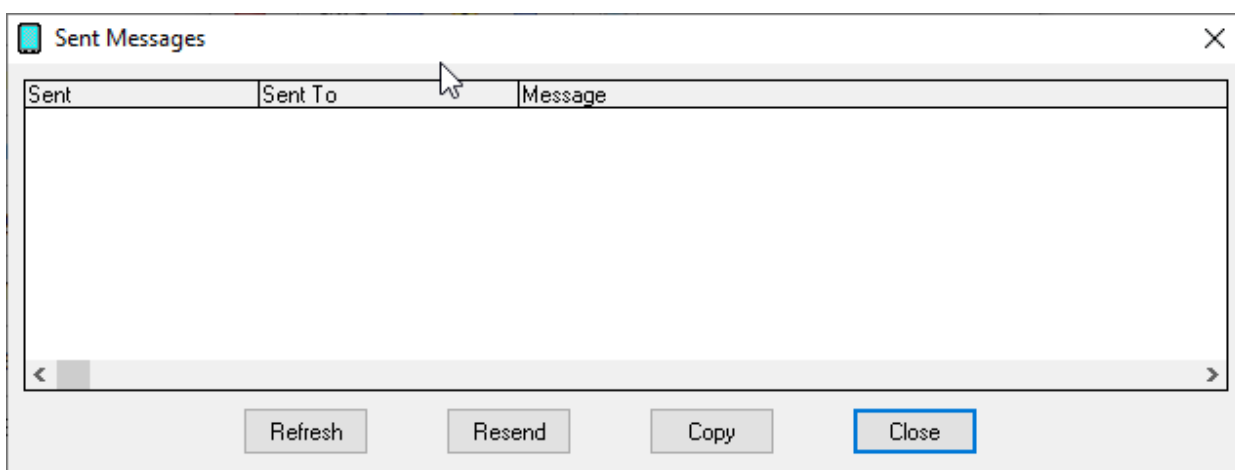
This option will display all messages sent from this Client's [station number](#) that are in the pending message queue.



Section	Function
Requested	This column displays the Date and Time at which the pending message was scheduled.
Sent To	This column displays to whom the message was intended.
Message	This column displays the message intended for the recipient or group.
Refresh	Clicking this button will refresh the list of Pending Messages.
Delete	Highlighting a pending message and clicking this button will delete the message from the queue, allowing you to cancel any pending message.
Copy	By highlighting a Pending message and clicking this button, you will be taken back to the main GUI Client interface, the message will be inserted into the Message section and the Selected Recipients will be moved.
Close	This button closes the Pending Messages view.

Sent

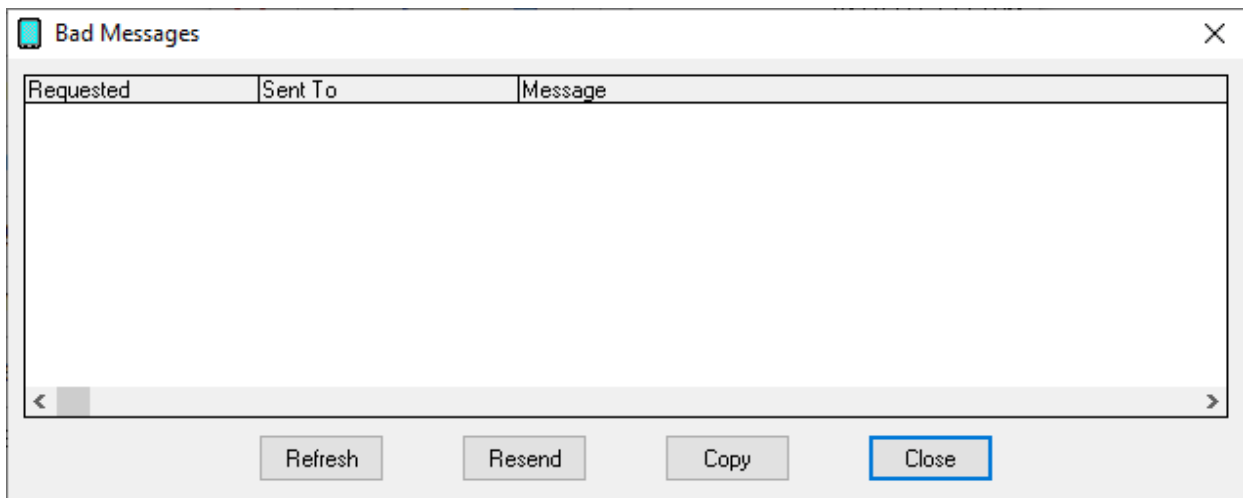
This option will display all messages sent from this Client's [station number](#) that were successfully delivered.



Section	Function
Sent	This column displays the Date and Time at which the message was sent.
Sent To	This column displays to whom the message was intended.
Message	This column displays the message sent to the recipient or group.
Refresh	Clicking this button will refresh the list of Sent Messages.
Requeue	Highlighting a failed message and clicking this button will queue this message to the originally specified recipients and groups for immediate delivery.
Copy	By highlighting a Sent message and clicking this button, you will be taken back to the main GUI Client interface, the message will be inserted into the Message section and the Selected Recipients will be moved.
Close	This button closes the Sent Messages view.

Bad

This option will display all messages sent from this Client's [station number](#) that could not be delivered.



Section	Function
Requested	This column displays the Date and Time at which the message failed.
Sent To	This column displays to whom the message was intended.
Message	This column displays the message sent to the recipient or group.
Refresh	Clicking this button will refresh the list of Sent Messages.
Requeue	Highlighting a failed message and clicking this button will queue this message to the originally specified recipients and groups for immediate delivery.

Copy	By highlighting a failed message and clicking this button, you will be taken back to the main GUI Client interface, the message will be inserted into the Message section and the Selected Recipients will be moved.
Close	This button closes the Bad Messages view.

Scheduled

This option will display all scheduled messages created by this Client's [station number](#).



Section	Function
Scheduled	This column displays the Date and Time at which the message was scheduled.
Sent To	This column displays to whom the message was intended.
Message	This column displays the message intended for the recipient or group.
Refresh	Clicking this button will refresh the list of Scheduled Messages.
Delete	Highlighting a scheduled message and clicking this button will delete the selected message.
Copy	By highlighting a scheduled message and clicking this button, you will be taken back to the main GUI Client interface, the message will be inserted into the Message section and the Selected Recipients will be moved.
Close	This button closes the Scheduled Messages view.

Replies

If two-way messaging has been configured and this workstation's sender name matches a local recipient on the PageGate server, this dialog displays the 'Replies' interface.



Section	Function
Sender	This column displays who or what sent the reply into the workstation.
Received	This column displays the date and time the message was received.
Message	This column displays the message received.
Refresh	Clicking this button will refresh the client's view to display the most recent information.
Resolved	Highlighting a message and clicking this button removes it from the list of messages.
Reply	Highlighting a message and clicking this button attempts to reply to a message. If a reply route does not exist, a warning message will inform the user that a reply can't be made.
Move	Highlighting a message and clicking this button allows a workstation to 'move' a reply to any recipient or group, whether that's another workstation or an actual recipient or group.
Copy	Highlighting a message and clicking this button allows a workstation to copy a message to any recipient or group, whether that's another workstation or an actual recipient or group.
Details	Highlighting a message and clicking this button displays a more detailed view of the message.
Close	Clicking this button closes the window.

History

This option will display the entire history of messages, good or bad, for this Client's [station number](#).



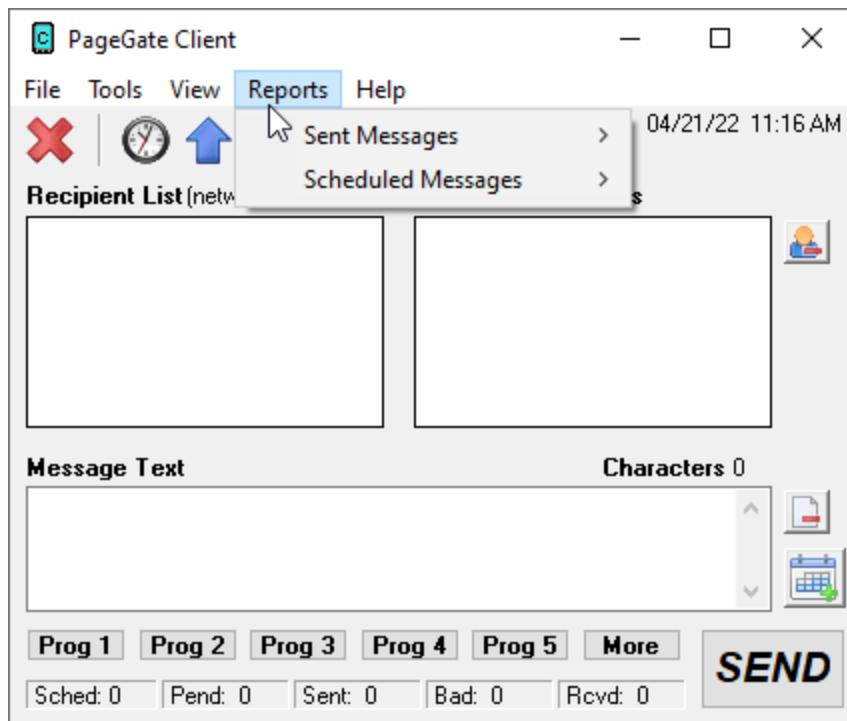
Section	Function
Requested	This column displays the Date and Time at which the message was scheduled.
Sent	This column displays the Date and Time at which the process to deliver the message was completed.
Status	This column displays whether the process to deliver the message was successful or if the message could not be delivered.
Sent To	This column displays to whom the message was intended.
Sent From	This column displays the Sender that queued the message.
Message	This column displays the message intended for the recipient.
Print	Clicking this button will send the Message History Log to the printer.
Clipboard	This button will copy the entire Message History Log to the clipboard and will allow you to paste the message into a text editor or email.
Resend	Highlighting any message and clicking this button will resend this message to the originally specified recipients and groups.
Copy	By highlighting any message and clicking this button, you will be taken back to the main GUI Client interface, the message will be inserted into the Message section

and the Selected Recipients will be moved.

Close This button closes the History Log view.

Reports

This section allows you to run reports on messages sent from any GUI Client that shares a [station number](#) with this Client.



- [Sent Messages](#)

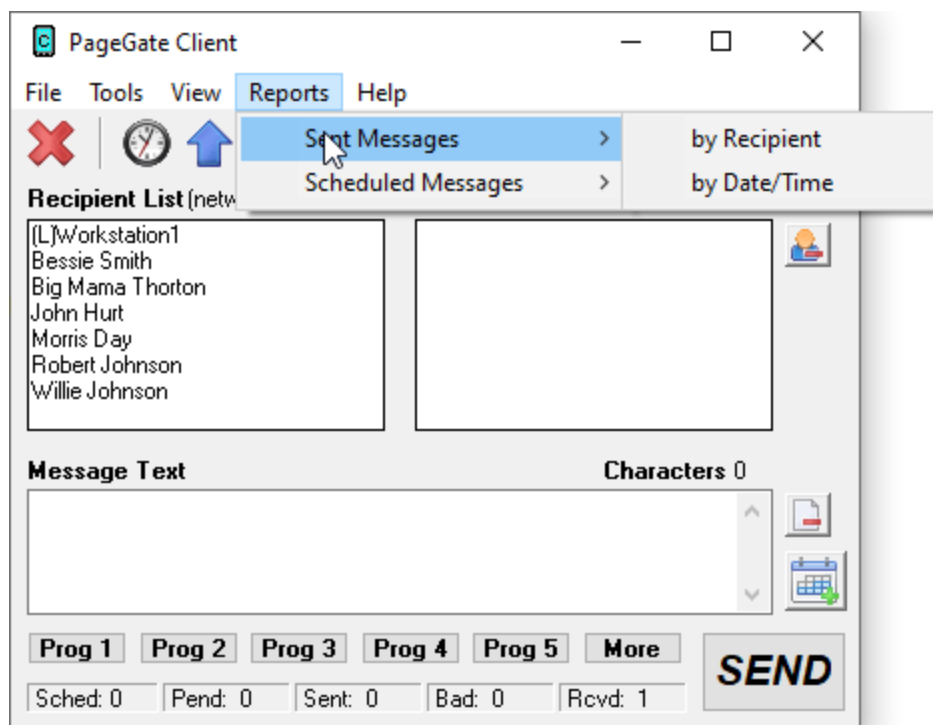
This section allows you to run reports on messages sent from any GUI Client that shares a [station number](#) with this Client.

- [Scheduled Messages](#)

This section allows you to run reports on scheduled messages created by any GUI Client that shares a [station number](#) with this Client.

Sent Messages

This section allows you to run reports on messages sent from any GUI Client that shares a [station number](#) with this Client.



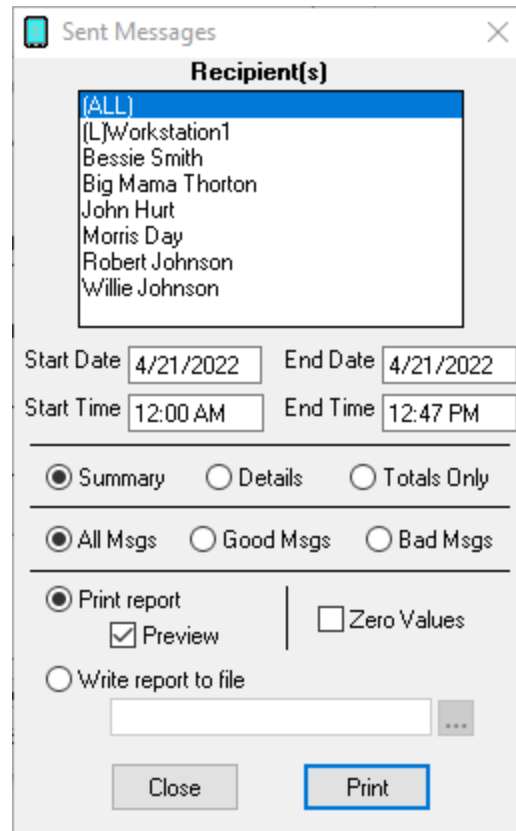
- [Sent Messages by Recipient](#)

This section allows you to run reports based on the recipient or group.

- [Sent Messages by Date/Time](#)

This section allows you to run reports on messages in a date and time range.

By Recipient



Setting	Function
Recipient(s)	You can select a single recipient or group or leave the (ALL) option highlighted to run a report on all recipients and groups. To select multiple recipients and groups, Ctrl+left click on each item you want to select.
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.
End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u>

	Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Zero Values	Enabling this option will include results that have a value of 0. This option is disabled by default.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

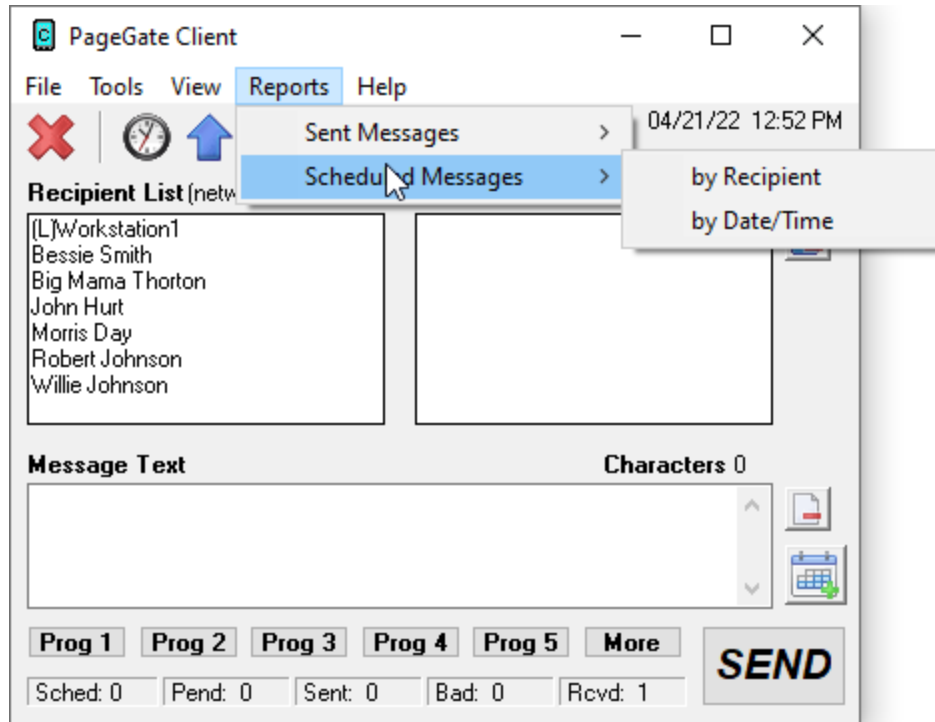
By Date/Time

Setting	Function
Start Date	The beginning date of the date range the report should show.
End Date	The ending date of the date range the report should show.
Start Time	The beginning time of the time range the report should show.

End Time	The beginning time of the time range the report should show.
Summary, Details, Totals Only	<ul style="list-style-type: none"> • <u>Summary</u> Selecting this option will give a brief overview of the information requested • <u>Details</u> Selecting this option will include all possible information requested. • <u>Totals Only</u> Selecting this option will only output the number of messages reported on without giving any information about the messages themselves.
All Msgs, Good Msgs, Bad Msgs	<ul style="list-style-type: none"> • <u>All Msgs</u> Selecting this option will run the report on all messages, good or bad. • <u>Good Msgs</u> Selecting this option will only report on messages that were delivered. • <u>Bad Msgs</u> Selecting this option will only report on messages that could not be delivered.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Scheduled Messages

This section allows you to run reports on scheduled messages created by any GUI Client that shares a [station number](#) with this Client.



- [Scheduled Messages by Recipient](#)

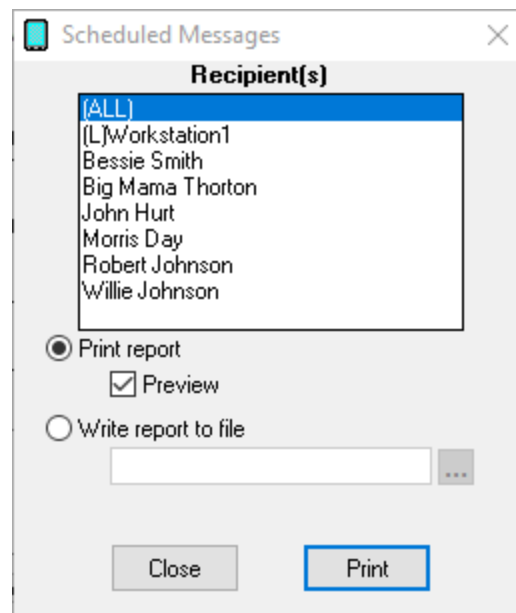
This section allows you to run reports based on the recipient or group.

- [Scheduled Messages by Date/Time](#)

This section allows you to run a report of all scheduled messages created by any GUI Client that shares a [station number](#) with this Client.

By Recipient

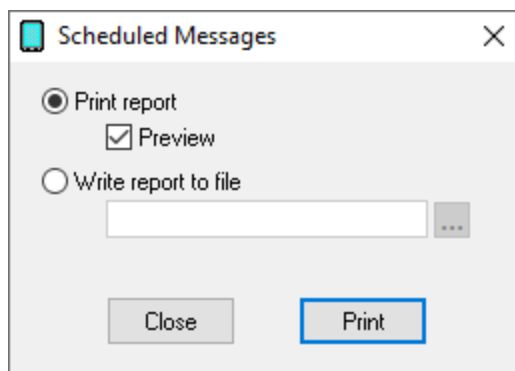
This section allows you to run reports based on the recipient or group.



Setting	Function
Recipient(s)	You can select a single recipient or group or leave the (ALL) option highlighted to run a report on all recipients and groups. To select multiple recipients and groups, Ctrl+left click on each item you want to select.
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

By Date/Time

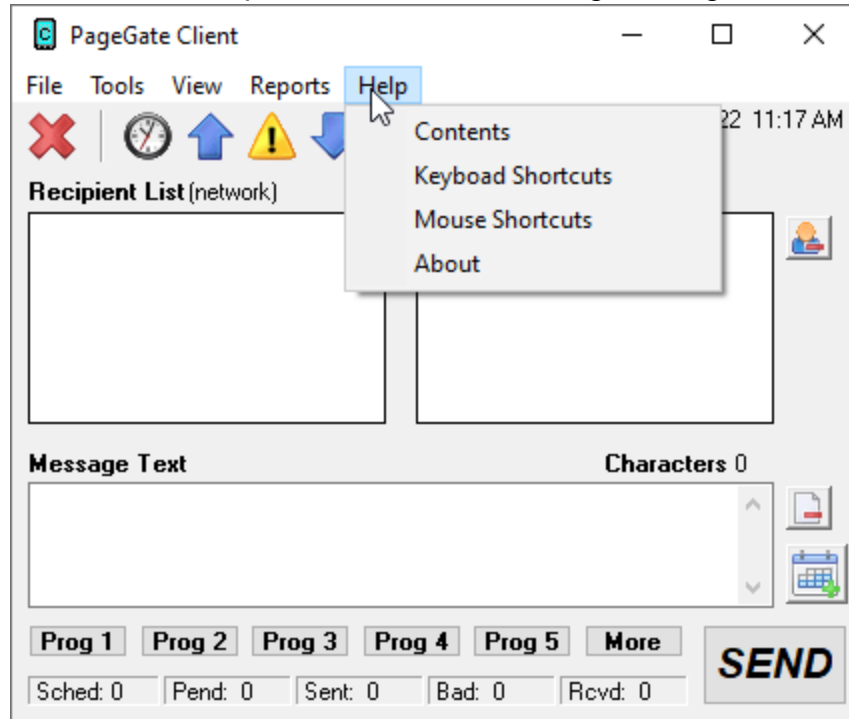
This section allows you to run a report of all scheduled messages created by any GUI Client that shares a [station number](#) with this Client.



Setting	Function
Print Report	This option allows you to print the report to a selected printer.
Preview	When this option is enabled, before the report is actually printed, you are shown the details of the report and can choose to either close the preview, in which case nothing is printed, or print the report.
Write report to file	When this option is enabled, the report is written to a file of your specification and name instead of queued to the printer.

Help

This section provides basic help and assistance for using the PageGate Client.



- [Contents](#)

This section will open the help file.

- [Keyboard Shortcuts](#)

This section will display all keyboard shortcuts that can be used with the PageGate Client.

- [Mouse Shortcuts](#)

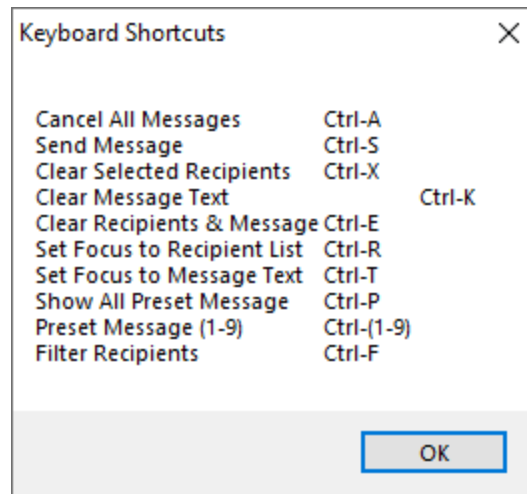
This section will display all mouse shortcuts that can be used with the PageGate Client.

- [About](#)

This section displays version and registration information for this installation of the PageGate Client.

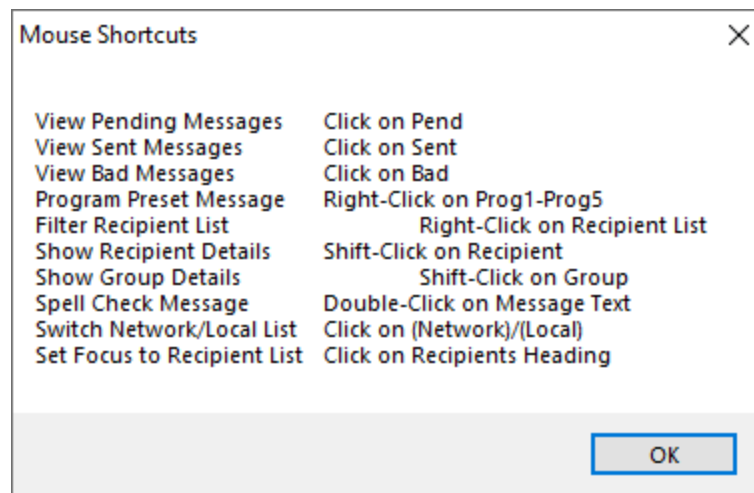
Keyboard Shortcuts

This section displays all keyboard shortcuts that can be used with the PageGate Client.



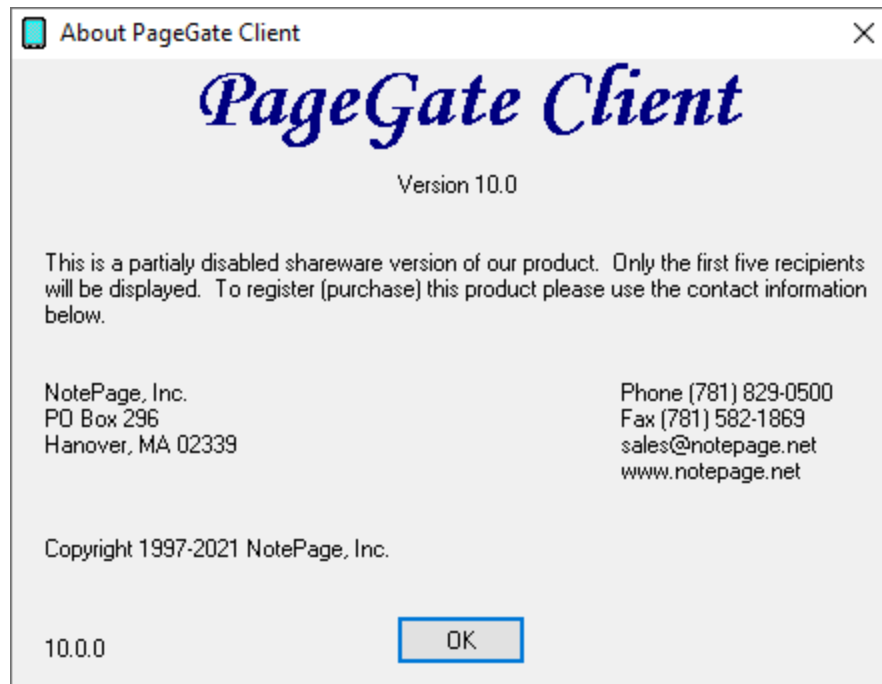
Mouse Shortcuts

This section displays all mouse shortcuts that can be used with the PageGate Client.

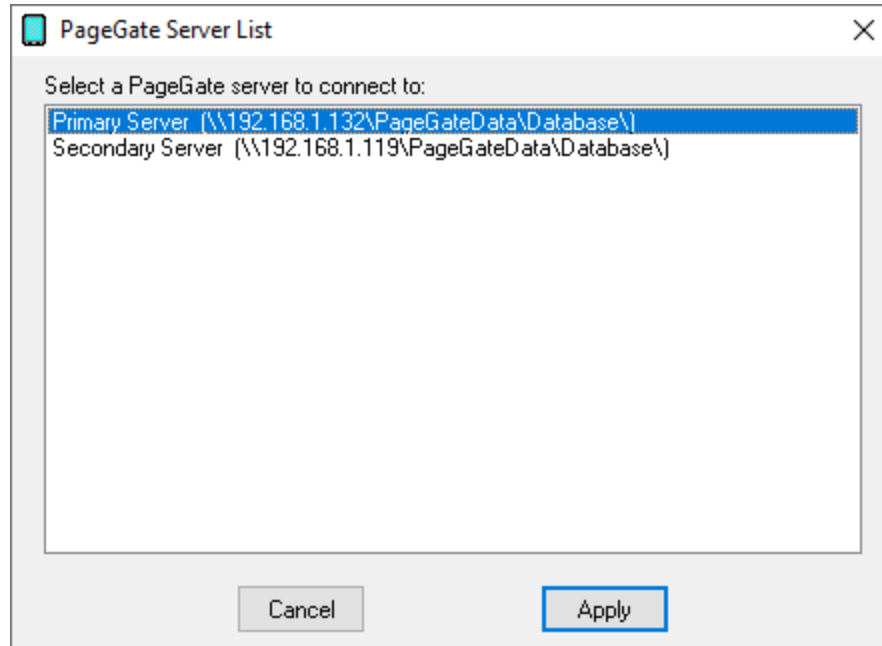


About

This section displays version and registration information.



Using a Single GUI Client with Multiple Servers



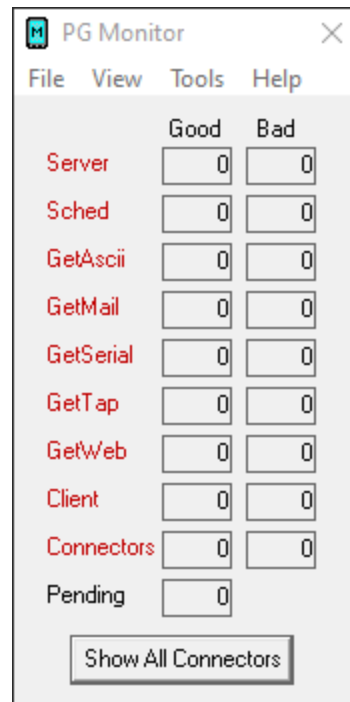
It is possible for the PageGate Client reference more than one PageGate server, though not simultaneously. If you provide the appropriate list of paths, the GUI Client will prompt the user for which server to connect to.

PageGate GUI Client

- 1) Run Notepad or your preferred text editor.
- 2) On each line of the file, specify the following:
 ServerName, UNC or Mapped Drive path to a PageGate server's database.
 Examples:
 Primary Server, \\192.168.1.132\PageGateData\Database\
 Secondary Server, \\192.168.1.119\PageGateData\Database\
- 3) Save the file in the PageGate Client program directory and name it 'serverlist'. You may want to verify that your text editor is saving the file with a .txt extension as the full file name needs to be 'serverlist.txt'.

If the file is present and formatted correctly, you'll receive a prompt for the server to connect to when the GUI Client is launched.

The PageGate Monitor (PG Monitor) is used for monitoring all aspects of the PageGate Server. It shows a set of real time log windows that display what each module of the program has done and is currently doing. The PageGate Monitor can also be configured to pass special alerts to NotePager Pro to notify someone when the PageGate server encounters an issue.



PG Monitor can be run from the PageGate server or on an administrator's workstation. Items shown in green are currently running. Items shown in red are stopped.

The Good and Bad Columns show how many messages each module has successfully or unsuccessfully processed.

The Pending category shows the number of messages currently waiting to be delivered.

- [File](#)

This section allows you to configure the basic settings of the PageGate Monitor as well as configure the NotePager Pro alerts function.

- [View](#)

This section allows you to select which logging windows will display.

- [Tools](#)

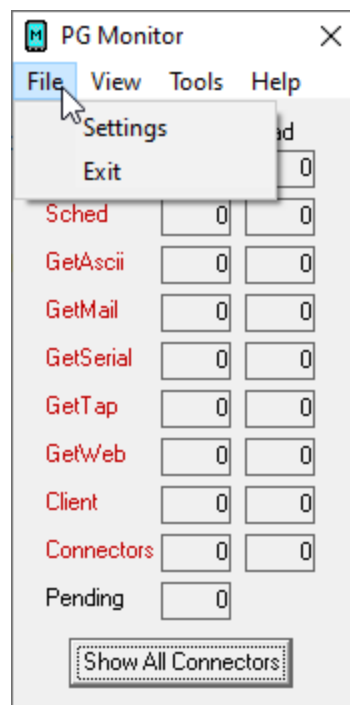
This section allows you to reset the counters or clear the visible logs from this instance of the PageGate Monitor.

- [Help](#)

This section provides basic help and tools for using the PageGate Monitor.

File

This section allows you to configure the basic settings of the PageGate Monitor as well as configure the NotePager Pro alerts function.



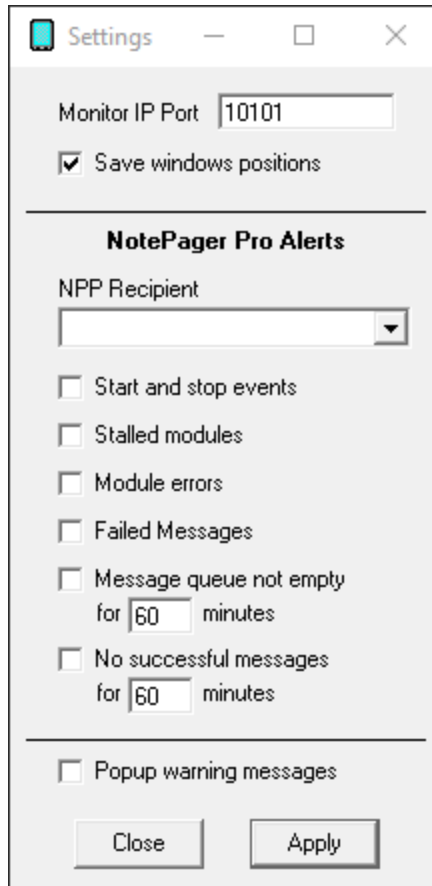
- [Settings](#)

This section allows you to configure the basic settings of the PageGate Monitor as well as configure the NotePager Pro alerts function.

- Exit

Selecting this option will close the PG Monitor.

Settings



Setting	Function
Monitor IP Port	This TCP/IP port on which the PageGate Monitor receives its data. This setting must match the port specified in the Logging Settings of the PageGate Admin.
Save windows positions	If you move or modify the position, shape, or size of the log windows displayed by the PageGate Monitor and don't have this option checked, the log windows will revert to their original state each time you close the Monitor.

NotePager Pro Alerts

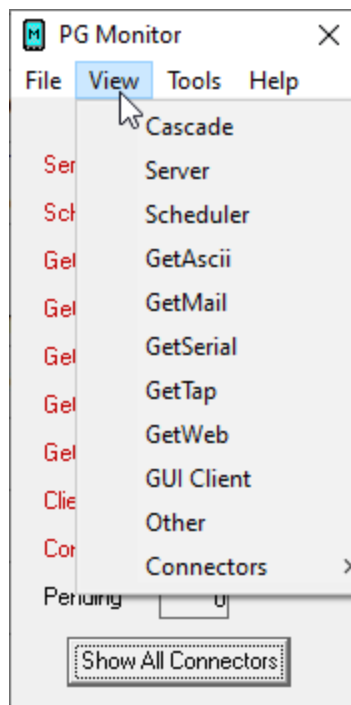
Note: Everything under this heading requires NotePager Pro to be installed and configured. For more information, please refer to our website:

<https://www.notepage.net/notepagerpro.htm>

NPP Recipient	Use this field to specify a recipient or group you've configured within NotePager Pro. All alerts sent from the PageGate Monitor will go to this NotePager Pro Recipient.
Start and stop events	When this option is enabled, all instances of a PageGate Module Starting or Stopping will be messaged to the specified NotePager Pro Recipient or Group.
Stalled modules	When this option is enabled, if a module has stalled or is unresponsive in Windows, the specified NotePager Pro Recipient or Group will be messaged.
Module errors	When this option is enabled, if a module produces an error message, that error message will be sent to the specified NotePager Pro Recipient or Group.
Failed Messages	When this option is enabled, if a message fails, the specified NotePager Pro Recipient or Group will be notified.
Message queue not empty for X minutes	When this option is enabled, if the number of pending messages never reaches zero for a specified amount of time, the specified NotePager Pro Recipient or Group will be notified.
No successful messages for X minutes	When this option is enabled, if no message is successfully sent for X number of minutes, the specified NotePager Pro Recipient or Group will be notified.
Close	This button will close the Settings of the PageGate Monitor. If you haven't Applied your changes, they will be lost.
Apply	This button will save all modifications you've recently made to the Settings of the PageGate Monitor.

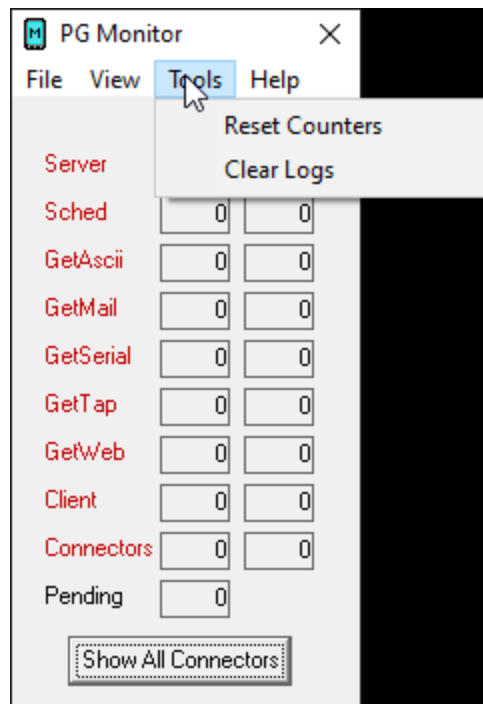
View

This section allows you to select which logging windows will display.



The View menu allows you to specify which log windows are shown. Generally, you should always run the Server, Scheduler, and Connector1 log windows; the rest are optional. You can specify whether a log window is shown by checking or unchecking each option under the View menu.

Tools



- Reset Counters

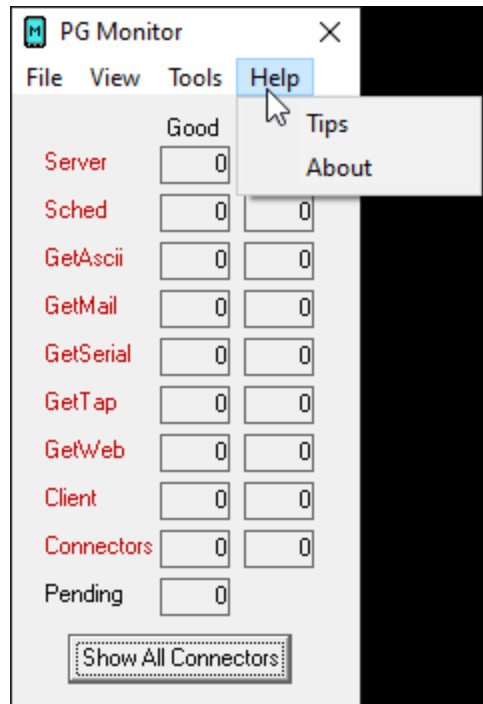
Selecting this option will reset the good and bad counters for this instance of the PG Monitor to 0.

- Clear Logs

Selecting this option will clear the visible log windows for this instance of the PG Monitor.

Help

This section provides basic help and tools for using the PageGate Monitor.



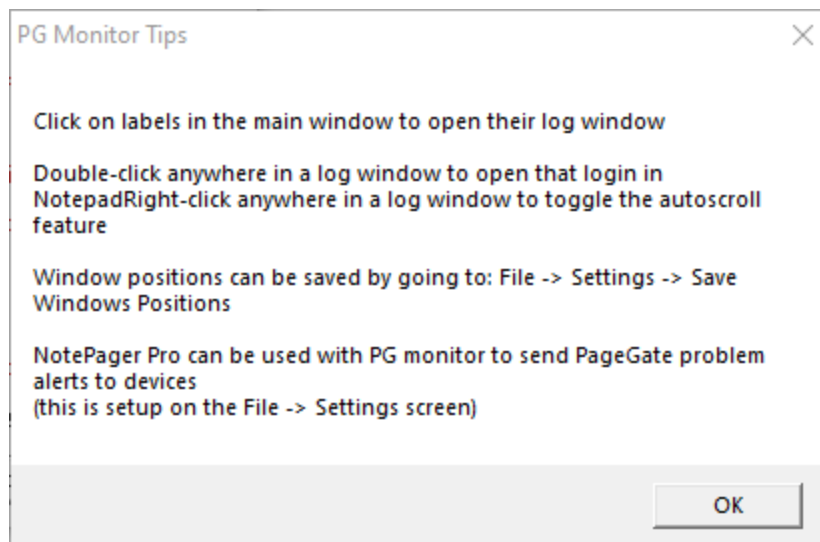
- [Tips](#)

This section will display a list of useful tips for using the PageGate Monitor.

- [About](#)

This section will display version information about this instance of the PageGate Monitor.

Tips



About

This section displays version information about this instance of the PageGate Monitor.



PageGate has a few key features that allow the program to monitor itself and potentially alert recipients and groups that there is an issue.

- [PageGate Monitor](#)

The PageGate Monitor (PG Monitor) is used for monitoring all aspects of the PageGate Server. It shows a set of real time log windows that display what each module of the program has done and is currently doing.

- [NotePager Pro alerts via PageGate Monitor](#)

Naturally, if PageGate is experiencing an issue, it may not be able to use its own messaging engine to notify you of the problem. The PageGate Monitor can be configured to send alerts through NotePager Pro's messaging engine in the event that certain alert criteria are met.

- [PageGate Log Monitoring](#)

PageGate's log monitor feature allows it to scan through its own log files and trigger certain behaviors, like executing a script and/or writing a file to a directory, based on your search criteria.

PageGate Log Monitor Feature

PageGate's log monitor feature allows the program to scan through its own log files and trigger certain behaviors based on what you're looking for.

Step 1: Create the LogMonitorInfo.txt file in the PageGate Database folder.

- 1) In Windows, browse in to your PageGate Database folder.
 - 1a) To determine its location, open PG Admin and go to Program - Settings
- 2) Create a new Text Document and name the file: LogMonitorInfo.txt
 - 2a) This file's name and location are very important and the file must be precisely named LogMonitorInfo.txt - make sure your text editor doesn't try to use a custom extension.
- 3) Open the file for editing.

Step 2: Determine what keywords and key phrases to scan for

Step 3: Determine which PageGate log(s) you need to scan

Step 4: Write the necessary lines in LogMonitorInfo.txt

This file is written in a Comma Separated Values format, which means each new line is a record and also that each object in each record will need to be separated by commas.

Each line of the file will reference 4 things and in this order:

Module,Match,Command,Continue

• **Module**

This will need to reference one of PageGate's executable names, without the extension.

For example, if you want to scan the global log, you would use pagegate. If you want to scan the scheduler log, you would use pgsched.

Again, this field needs to reference the file name of one of PageGate's executables, without the extension, and determines which log is being scanned.

PGDialX can be used as a module name to apply to all Connector modules.

• **Match**

This field is case sensitive and determines what text to search for in the scanned log.

Quotes, commas and backslashes in search terms will need to be expressed as their ASCII value, prefixed by a backslash (\).

Character	ASCII Value	Expression
\	092	\092
"	034	\034
,	044	\044

• **Command**

This determines what command should be executed if a match is found and can reference anything that can be called on from a command line. You could run cmd and echo a text file to a specific location as easily as you could execute a powershell script or run a batch file.

Keeping an eye on PageGate

The following macros are available for use in the Command value:

- %DateTime% - date and time of log message
- %Module% - the PG module name
- %Message% - the message being logged

- **Continue**

This value can be set to True or False.

If set to to True, successive lines in the file will be used.

If set to False, this line will be treated as the last relevant line in the file.

Step 5: Cycle the PageGate service to implement new settings

Changes to LogMonitorInfo.txt are not recognized in real time and require a service restart of PageGate's connectors. The easiest way to do this is:

- 1) Open PG Admin on the PageGate server. This cannot be done from a workstation Admin.
- 2) Right click Connectors and select 'Stop All'.
- 3) Right click Connectors and select 'Start All'.

Examples

This is an example that would scan the Connector 1 log file for any failed messages and echo all failed message log lines to c:\temp\failures.txt:

```
"PGDial1","FAILED", "cmd /c @echo %DateTime% %Module% %Message% >> c:\092temp\092failures.txt", "True"
```

This is an example that would scan all Connector for any instance of traffic being put through PageGate's failover system and execute a batch file that could, among other things, call on a Powershell script:

```
"PGDialX","Carrier retry limit reached. Redirecting messages", "c:\PageGateData\Scripts\YourBatchFile.bat", "True"
```

PG Monitor Alerts through NotePager Pro

This should not be done on the PageGate server.

Naturally, if PageGate is experiencing an issue, it may not be able to use its own messaging engine to notify you of the problem. The PageGate Monitor can be configured

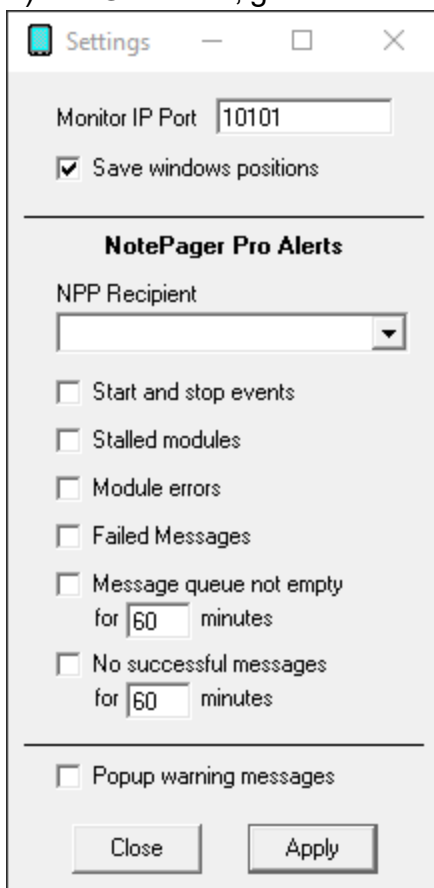
to send alerts through NotePager Pro's messaging engine in the event that certain alert criteria are met.

Step 1: Install the workstation PG Monitor and NotePager Pro on a workstation

- 1) Install the [workstation PG Monitor](#)
- 2) Install [NotePager Pro](#)
- 3) Configure at least one recipient and/or group in NotePager Pro. This will be the person or group of people who will receive any alerts.

Step 2: Configure the PG Monitor alerts

- 1) In PG Monitor, go to File - Settings



- 2) In the 'NPP Recipient' dropdown, select the recipient or group that should receive the alerts
- 3) Select the events that should trigger an alert:

Start and stop events	When this option is enabled, all instances of a PageGate Module Starting or Stopping will be messaged to the specified NotePager Pro Recipient or Group.
Stalled modules	When this option is enabled, if a module has stalled or is unresponsive in Windows, the specified NotePager Pro Recipient or Group will be

	messaging.
Module errors	When this option is enabled, if a module produces an error message, that error message will be sent to the specified NotePager Pro Recipient or Group.
Failed Messages	When this option is enabled, if a message fails, the specified NotePager Pro Recipient or Group will be notified.
Message queue not empty for X minutes	When this option is enabled, if the number of pending messages never reaches zero for a specified amount of time, the specified NotePager Pro Recipient or Group will be notified.
No successful messages for X minutes	When this option is enabled, if no message is successfully sent for X number of minutes, the specified NotePager Pro Recipient or Group will be notified.

4) Click Apply.

There are several places to find help on the operation of PageGate. This Help file is one of the best resource for most questions; it can be accessed from the PageGate program group in your Start Menu, or by clicking Help in the PageGate Admin.

Support can also be reached through the NotePage Forum at <https://www.notepage.net/forum/>

For more information, please visit the support section of our website: <https://www.notepage.net/support.htm>

NotePage Support is available Monday through Friday from 9:00 AM to 5:00 PM Eastern Time and can be reached by phone at 781-829-0500 x 2 or by email at support@notepage.com

Calls are handled on a first come, first served basis unless you have a priority support service license agreement. Priority support puts your calls and email in to a priority queue for faster response times and comes with many additional benefits.

For more information, please have a look at the priority section of our website at <https://www.notepage.net/priority-support.htm>

When contacting NotePage's Technical Support Department, please obtain the relevant PageGate Log files and database.

For more information on how to do that, please review the [Database](#) and [Logging](#) sections of the documentation.

All of PageGate's manuals are also available on in the support section of NotePage's website in HTML, PDF and Rich Text format. (found under the 'Support' section):
<https://www.notepage.net/support.htm>

Index

D

Delimiter 309

E

Email 307

F

From 307

L

Length 310

P

POP Collect Port 710

S

Search 310

Security Alias 274

Security Manager 636, 653

Sender 307

Subject 309, 310