Swift Email Verifier API Client Manual

Why use SEV API Client

SEV API Client is a simple and powerful JAVA based tool which is highly optimized to validate and scrub emails at speeds exceeding 1000 simultaneous connections. You do not need port 25 (SMTP port) to be open in your network in order to use the JAVA verifier. All you need is a good and stable internet connection. Please be advised that this JAVA email validation API client is suitable and optimized for validating moderate mailing list files containing less than 1 million email addresses in a single batch. If your mailing list is more than 1 million, it is advisable to break the list into several parts and verify them in multiple batches.

With SEV API Client you can easily import mailing lists containing email addresses and other custom fields or information such as names, addresses, phone numbers, zip codes etc for verification from CSV files which means the email verifier will ensure that you don't lose any contact information, such as first name, last name, company, phone, address, etc. It also provides you options to download emails according to various categories like valid, invalid, unknown and full results including any custom information contained in the original mailing list.

SEV API Client is totally fault tolerant and can take care of network failures and helps you save the results in between. After running the validation jobs, the results can be downloaded without internet connection.

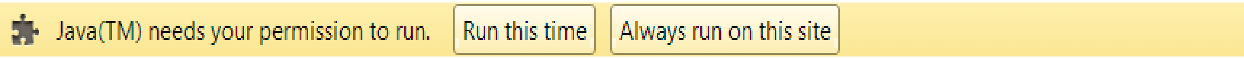
Pre-requisites for SEV API Client

Java Runtime Environment must be installed on your computer.

If you are not sure about the availability of Java on your system, please visit the link .” http://www.java.com/en/download/installed.jsp”

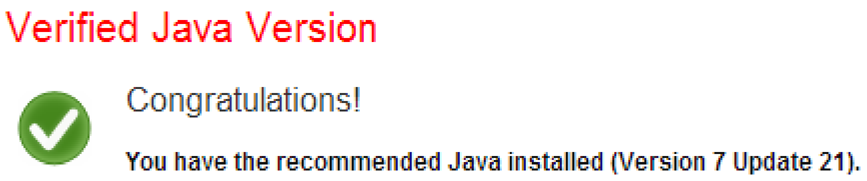
Click on the button which says “V**erify Java version”** as shown below:

You might get a warning which says Java™ needs your permission to run as below:


Java(TM) needs your permission to run.

Select **Run this time.**

If a message is shown as below, it means you have java installed on the system.



Verified Java Version

Congratulations!

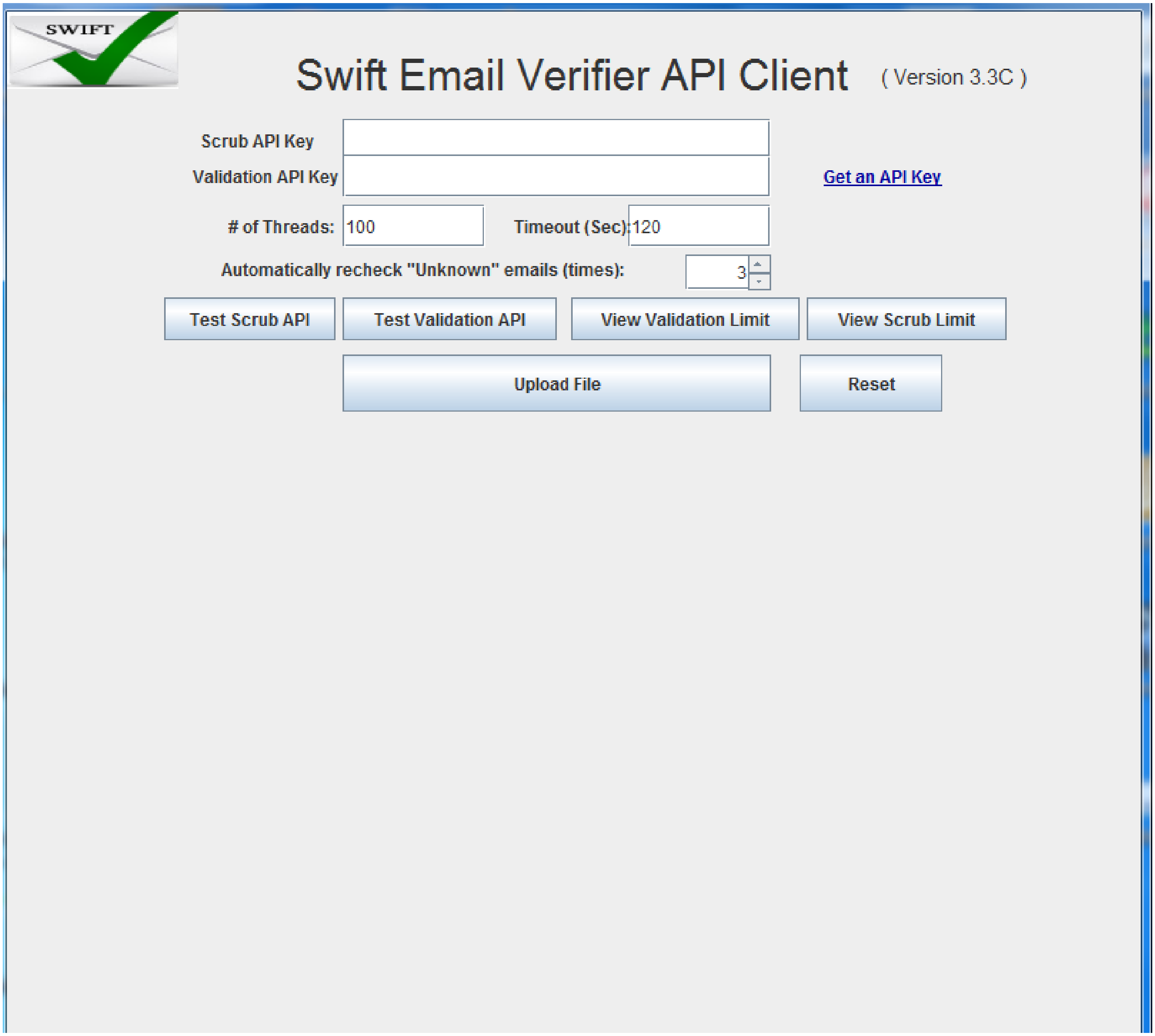
You have the recommended Java installed (Version 7 Update 21).

Operating System Requirements

Swift Email Verifier API Client is based on JAVA therefore it works on all operating systems platforms such as Windows, MAC and Linux. The application does not require any complex installation. Simply download and run and you are ready to start validating your emails.

Starting SEV API Client

Double click the jar file and it will open up the below main screen:



Swift Email Verifier API Client (Version 3.3C)

Scrub API Key

Validation API Key

# of Threads: 100

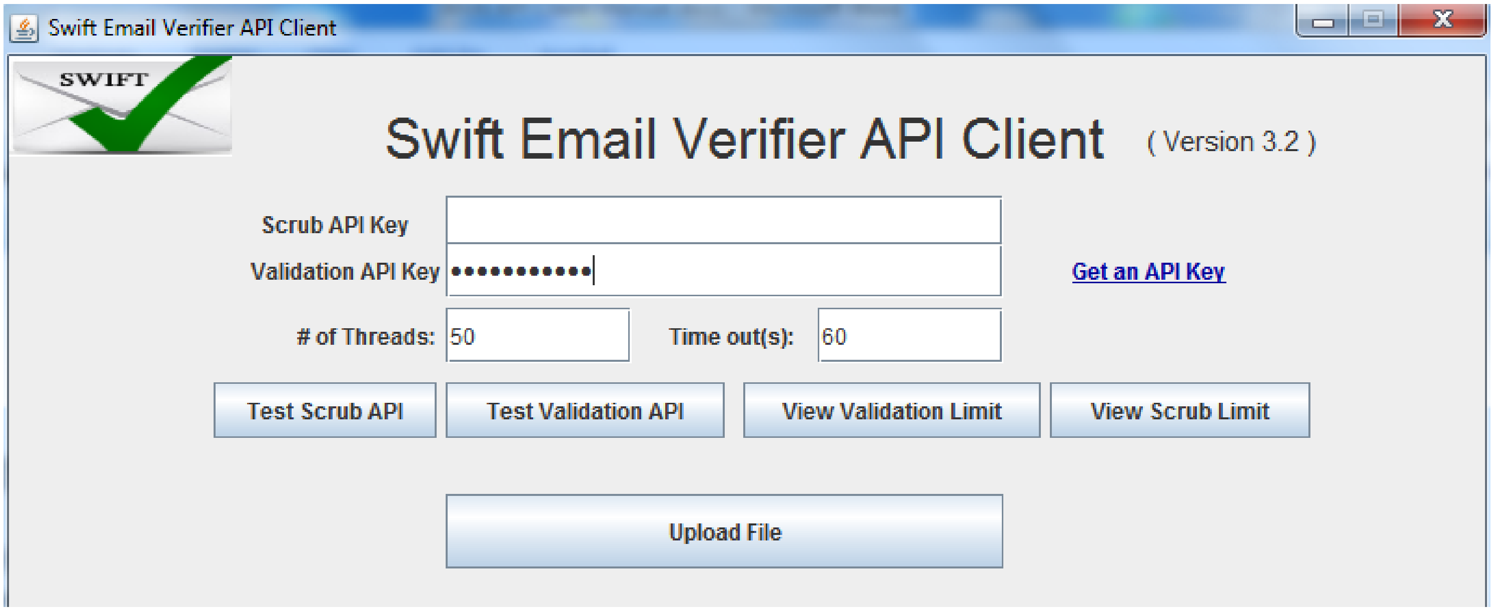
Timeout (Sec):120

Automatically recheck "Unknown" emails (times): 3

Using SEV API Client

Testing for API connectivity:

Type the email validation or email scrubbing API key in the appropriate text box and click the Test API button.



Swift Email Verifier API Client

Swift Email Verifier API Client (Version 3.2)

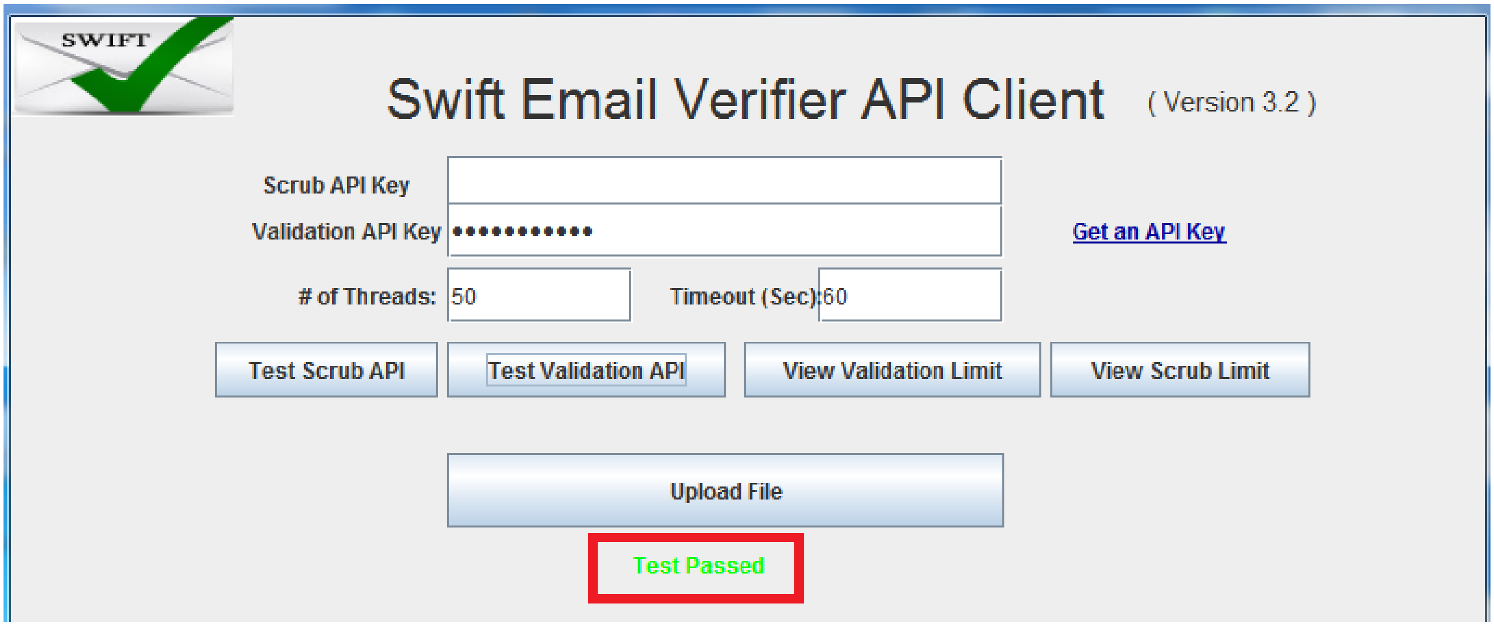
Scrub API Key:

Validation API Key: •••••••••••

# of Threads: 50

Time out(s): 60

The screen will show a success or a failure message depending upon the result of the test.



Swift Email Verifier API Client (Version 3.2)

Scrub API Key:

Validation API Key: •••••••••••

# of Threads: 50

Timeout (Sec): 60

Test Passed

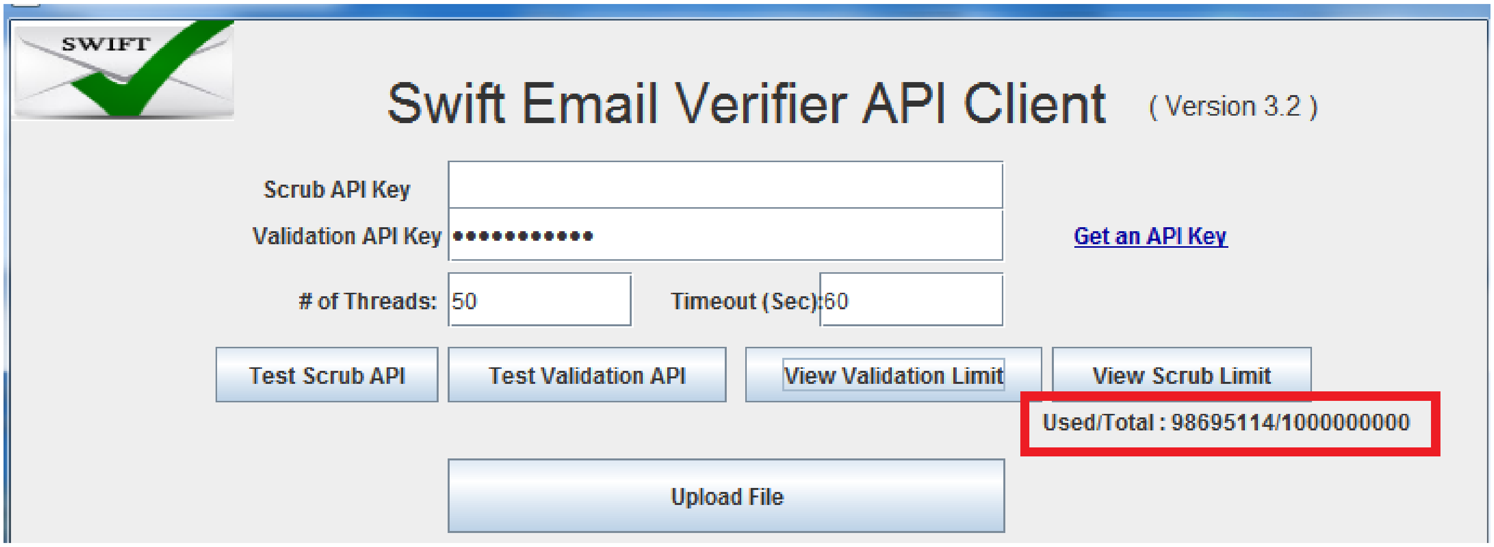
It is always recommended to test before starting a validation.

Note: You may have to click on the Test API button several times before receiving a “Test Passed” message. If the API key is valid and you continue to receive failed message for the key, please recheck the key again and make sure you are connected to the internet or contact us if you are certain that the API key is valid.

Checking Remaining Credits:

Type the Validation or Scrubbing API key in the appropriate text box and click the View Limit button.

It might take a while to appear and you will see the available limits adjacent to the button in the format ***“Used/Total: XXXX/YYYY”*** *for the email validation API or* ***“UsageLeft:XXXX”*** for the scrubbing API as shown in the screenshots below:

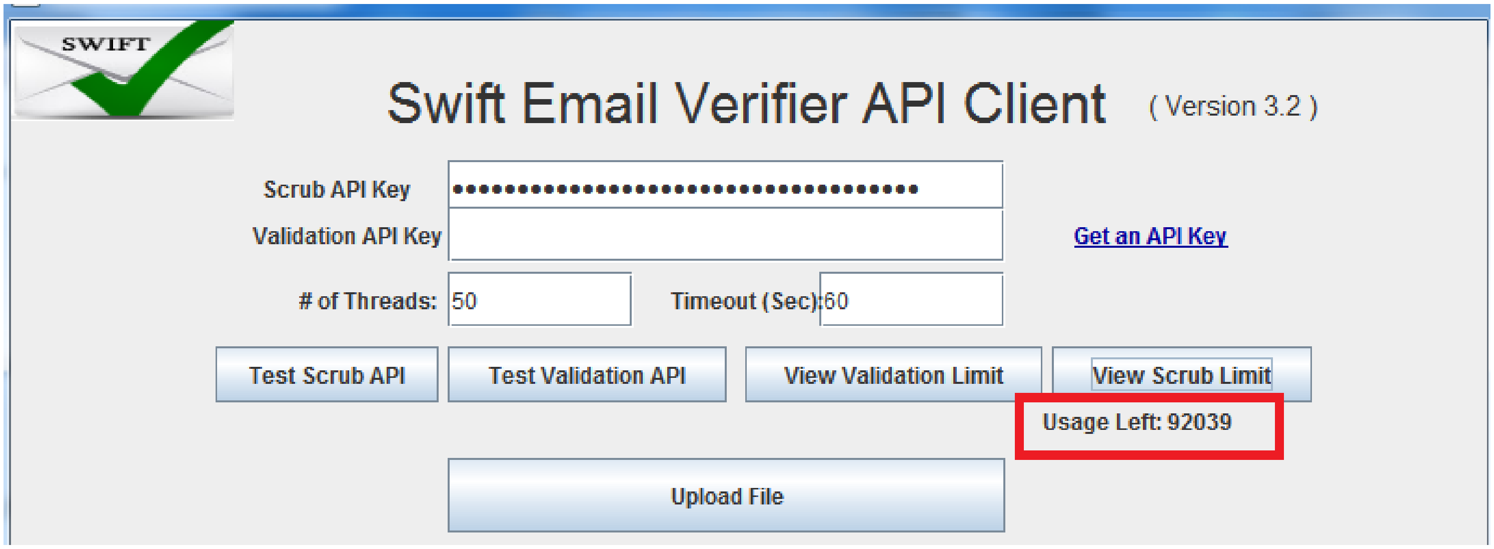


Validation API Key: ●●●●●●●●●●●

# of Threads: 50

Timeout (Sec):60

Used/Total : 98695114/1000000000



Scrub API Key: ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

# of Threads: 50

Timeout (Sec):60

Usage Left: 92039

Please always view limits before initiating the Validation to avoid unnecessary waste of time.

What is Checked by Email Validation API:

- ✓ **Email syntax:** This checks the email addresses syntax and ensures that they conforms to IETF standards
- ✓ **Fake Email Pattern Detection:** This checks the email address against a powerful in-built fake email pattern detector algorithm. This fake email pattern detector is capable of detecting thousands of fake email automatically with very high accuracy.
- ✓ **Typo Check and Curse Words Check:** This checks the email address against all known common typos for most email domains. The API can also detect certain curse words present in the email address.

✓**Mail Server Existence Check:** This checks the availability of the email address domain using DNS MX records

✓**Mail Existence Check:** This checks if the email address really exists and can receive email at that instant

✓**Catch-All Domain Email Check:** This checks if the email domain will receive all of the email messages addressed to that domain, even if their addresses do not exist in the mail server.

- ✓ **Disposable Email Address Check:** This checks if the email is provided by a known Disposable Email Address (DEA) provider such as Mailinator, 10MinuteMail, GuerrillaMail and about 2000 more.

WHAT IS REQUIRED TO USE THE PROGRAM:

To validate or scrub your email addresses using the application, you will need the following:
1. Your Email validation API Key or email scrubbing API key (APIKey)
2. The mailing list in the proper and supported format.

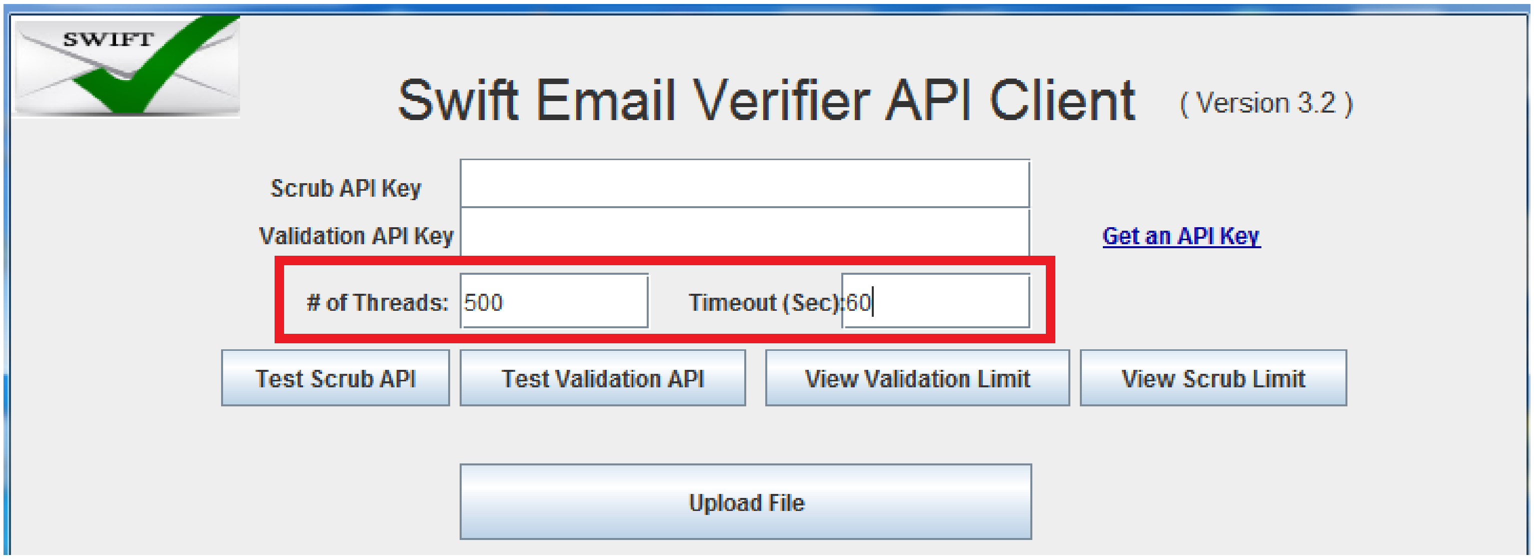
API KEY AUTHENTICATION and RESTRICTIONS:

Clients must authenticate to the API by providing their API key. Care must be taken to secure the key from unauthorized access. It is your responsibility to keep your API key secure at all times and ensure that unauthorized users do not have access to it.

The API keys can be top-up at any time and will remain valid until all credits have been used up. The API key can also be used by multiple persons from unlimited devices or computers at the same time without any restrictions.

Configuring the number of Simultaneous Connections (Threads) & Timeout:

The application is a multithreaded program that is capable of achieving over 1000 simultaneous connections per second. To ensure your email validations are run faster, enter your desired # of connections for multi-threaded validations. To speed up your email validations, you should enter your desired number of parallel connections in the "# of Threads" field and an appropriate timeout in program as illustrated in the screenshot below. Our API can support up to 1000 instantaneous connections depending on the speed of your internet connection and computer specifications such as RAM and CPU.



Scrub API Key

Validation API Key

# of Threads: 500

Timeout (Sec): 60

Please note that if you set the # of connections too high with a low timeout, you will encounter the error: "InternalError" To avoid this error, you should set the timeout to a higher value such as 120 or 180 seconds which has been tested to be ideal for good network connections.

If your network speed is slow, you may have to increase the timeout to a higher number such as 120 or 200 seconds. In addition, to prevent connection timeout errors, it is advisable to add the program to the list of allowed programs in your firewall settings if using one in your computer. If you have anti-virus software, please switch it off before verifying the email addresses.

Configuring the number of Automatic Re-Check of Unknown:

Due to multiple factors such as network issues, rare server outrage issues or inability to verify an email address where the ISP do not cooperate with the email validation method because it requires an actual message to be sent, unknown results are bound to happen when using our API.

However, since a majority of these network issues causing the unknown results are transient (temporary) it makes sense to retry the emails again. To this end , the latest version of SEV has a feature to automatically re-check or re-validate emails up to a specified number of times in order to improve the success of the validations and minimize unknowns as much as possible.

To configure the number of times you want the application to automatically re-check an email address which previously gave an unknown result, go to the “Automatically re-check unknown emails “ setting and enter a number there. The default value which is 3 is quite OK for most network conditions.

Validating Emails:

This is a multi-step process as described in the following steps below.

Step 1: Upload Mailing List File

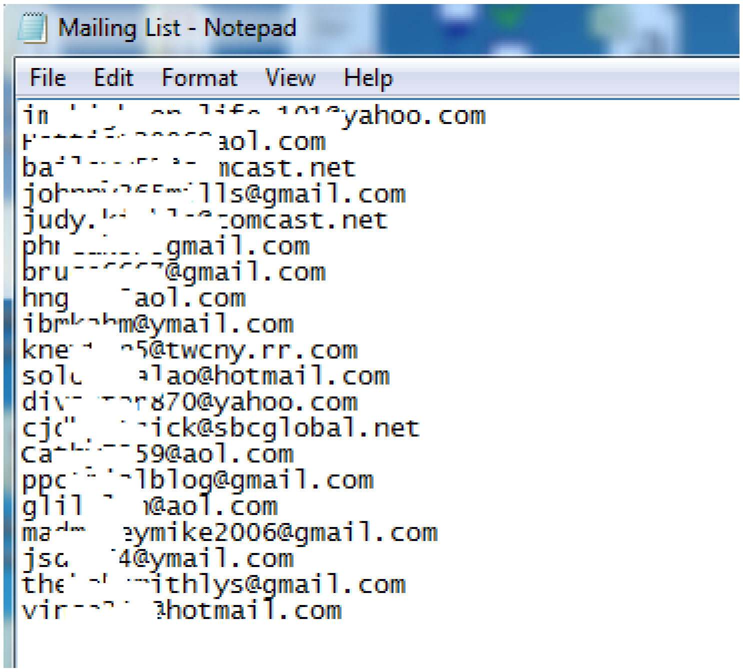
Supported Mailing List Formats:

You can upload csv or txt format files to add mailing list to the program. Swift Email Verifier API client only supports 2 types of mailing list file formats as follows:

- TEXT (.txt)
- CSV (.csv)

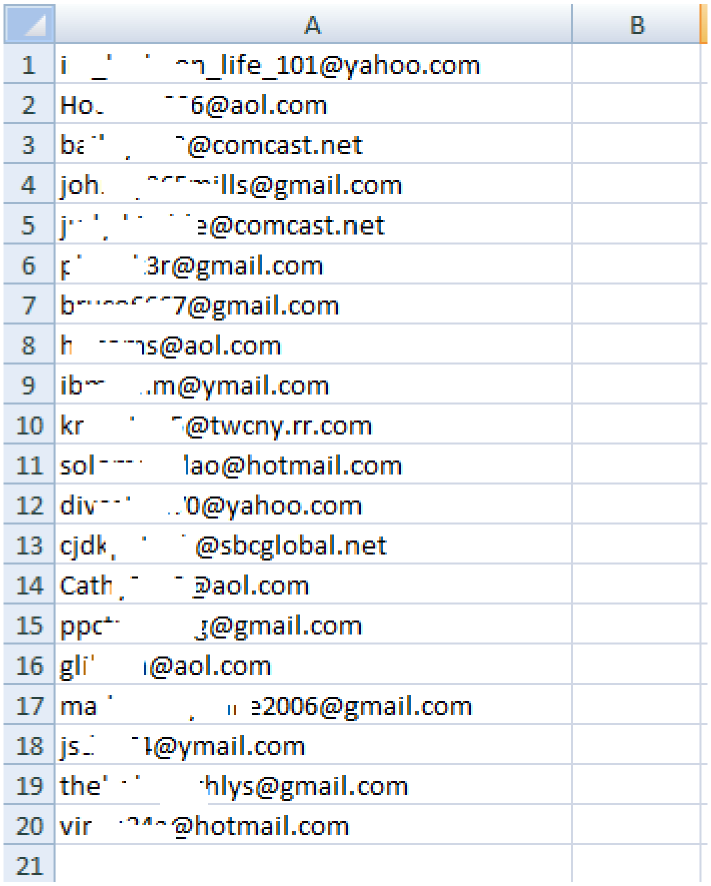
The mailing list can be uploaded in either .txt or .csv formats. Custom fields or information such as names, zip codes, addresses or phone numbers are supported and may be present in the mailing lists. If the mailing lists contain extra information, the validation results will also retain the extra information. Note that when you upload your mailing lists into the program, duplicates are automatically removed. This ensures that all email addresses imported into the program is unique.

Samples screenshots for the mailing lists in both TXT and CSV formats are shown below:



```
in[redacted]yahoo.com
[redacted]aol.com
ba[redacted]mcast.net
joh[redacted]lls@gmail.com
judy.[redacted]omcast.net
ph[redacted]gmail.com
bru[redacted]@gmail.com
hng[redacted]aol.com
ibr[redacted]m@ymail.com
kne[redacted]5@twcny.rr.com
sol[redacted]lao@hotmail.com
div[redacted]870@yahoo.com
cj[redacted]ick@sbcglobal.net
ca[redacted]59@aol.com
ppc[redacted]lblog@gmail.com
glil[redacted]@aol.com
ma[redacted]eymike2006@gmail.com
js[redacted]4@ymail.com
the[redacted]ithlys@gmail.com
vir[redacted]hotmail.com
```

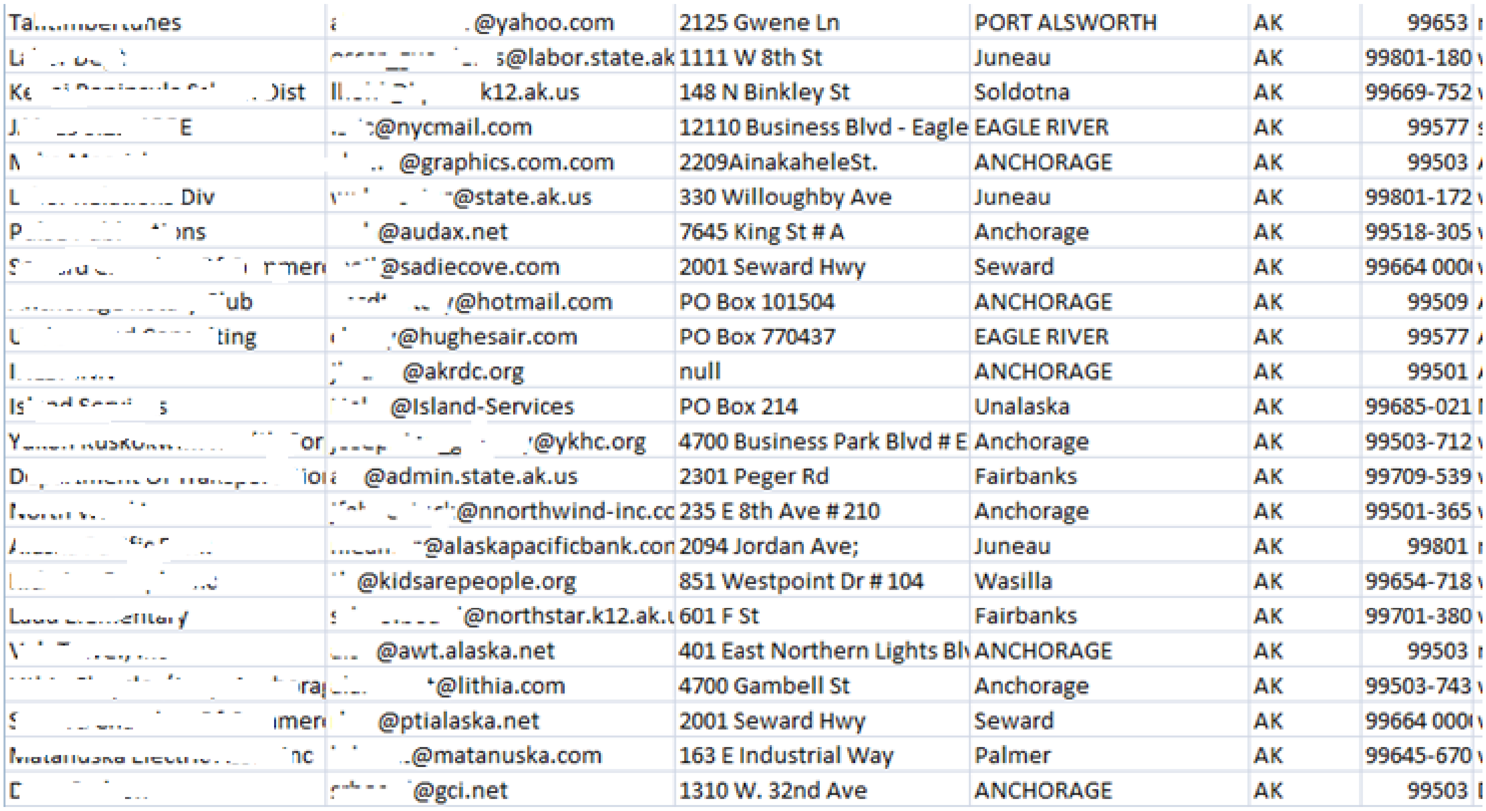
Mailing List in .txt format



	A	B
1	i _ n_life_101@yahoo.com	
2	Ho 6@aol.com	
3	ba @comcast.net	
4	joh ills@gmail.com	
5	j e@comcast.net	
6	p 3r@gmail.com	
7	b 7@gmail.com	
8	h ns@aol.com	
9	ib m@ymail.com	
10	kr @twcny.rr.com	
11	sol lao@hotmail.com	
12	div 0@yahoo.com	
13	cjdk @sbcglobal.net	
14	Cath @aol.com	
15	ppc g@gmail.com	
16	gli @aol.com	
17	ma e2006@gmail.com	
18	js 1@ymail.com	
19	the hlys@gmail.com	
20	vir @hotmail.com	
21		

Mailing list in .csv format

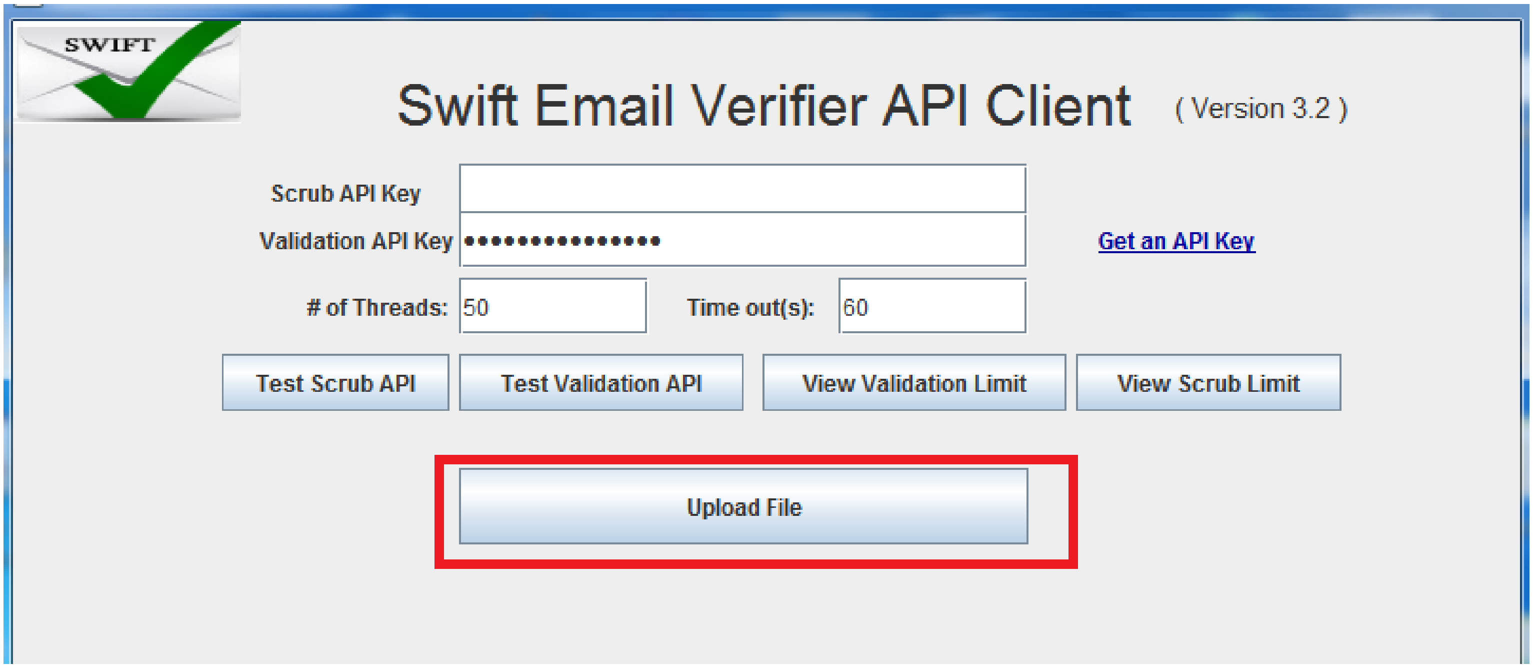
Swift Email Verifier API Java Client- User Manual



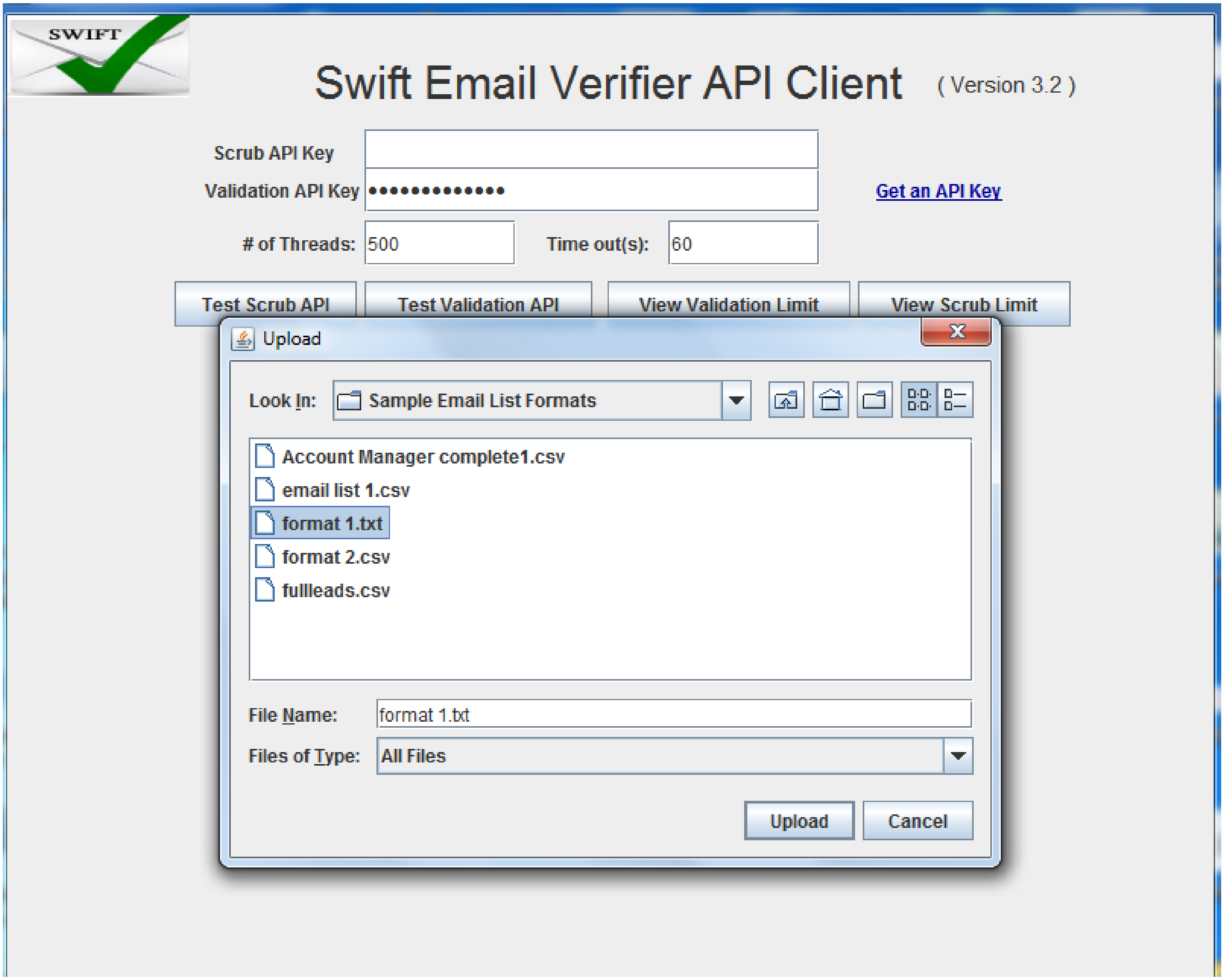
[redacted]	[redacted]@yahoo.com	2125 Gwene Ln	PORT ALSWORTH	AK	99653
[redacted]	[redacted]s@labor.state.ak	1111 W 8th St	Juneau	AK	99801-180
[redacted] Dist	[redacted]k12.ak.us	148 N Binkley St	Soldotna	AK	99669-752
[redacted]E	[redacted]@nycmail.com	12110 Business Blvd - Eagle	EAGLE RIVER	AK	99577
[redacted]	[redacted]@graphics.com.com	2209AinakaheleSt.	ANCHORAGE	AK	99503
[redacted] Div	[redacted]@state.ak.us	330 Willoughby Ave	Juneau	AK	99801-172
[redacted]ns	[redacted]@audax.net	7645 King St # A	Anchorage	AK	99518-305
[redacted]mer	[redacted]@sadiecove.com	2001 Seward Hwy	Seward	AK	99664 000
[redacted]ub	[redacted]@hotmail.com	PO Box 101504	ANCHORAGE	AK	99509
[redacted]ting	[redacted]@hughesair.com	PO Box 770437	EAGLE RIVER	AK	99577
[redacted]	[redacted]@akrdc.org	null	ANCHORAGE	AK	99501
[redacted]	[redacted]@Island-Services	PO Box 214	Unalaska	AK	99685-021
[redacted]or	[redacted]@ykhc.org	4700 Business Park Blvd # E	Anchorage	AK	99503-712
[redacted]ion	[redacted]@admin.state.ak.us	2301 Peger Rd	Fairbanks	AK	99709-539
[redacted]	[redacted]@nnorthwind-inc.cc	235 E 8th Ave # 210	Anchorage	AK	99501-365
[redacted]	[redacted]@alaskapacificbank.con	2094 Jordan Ave;	Juneau	AK	99801
[redacted]	[redacted]@kidsarepeople.org	851 Westpoint Dr # 104	Wasilla	AK	99654-718
[redacted]entary	[redacted]@northstar.k12.ak.u	601 F St	Fairbanks	AK	99701-380
[redacted]	[redacted]@awt.alaska.net	401 East Northern Lights Bl	ANCHORAGE	AK	99503
[redacted]ra	[redacted]@lithia.com	4700 Gambell St	Anchorage	AK	99503-743
[redacted]mer	[redacted]@ptialaska.net	2001 Seward Hwy	Seward	AK	99664 000
[redacted]nc	[redacted]@matanuska.com	163 E Industrial Way	Palmer	AK	99645-670
[redacted]	[redacted]@gci.net	1310 W. 32nd Ave	ANCHORAGE	AK	99503

Mailing list in .csv format with extra information

To begin uploading the mailing list file into the program, click on the Upload File button and choose the file to be uploaded.



Scrub API Key:
Validation API Key: ●●●●●●●●●●●●●●●
# of Threads: 50
Time out(s): 60



Scrub API Key

Validation API Key ••••••••••••••

# of Threads: 500

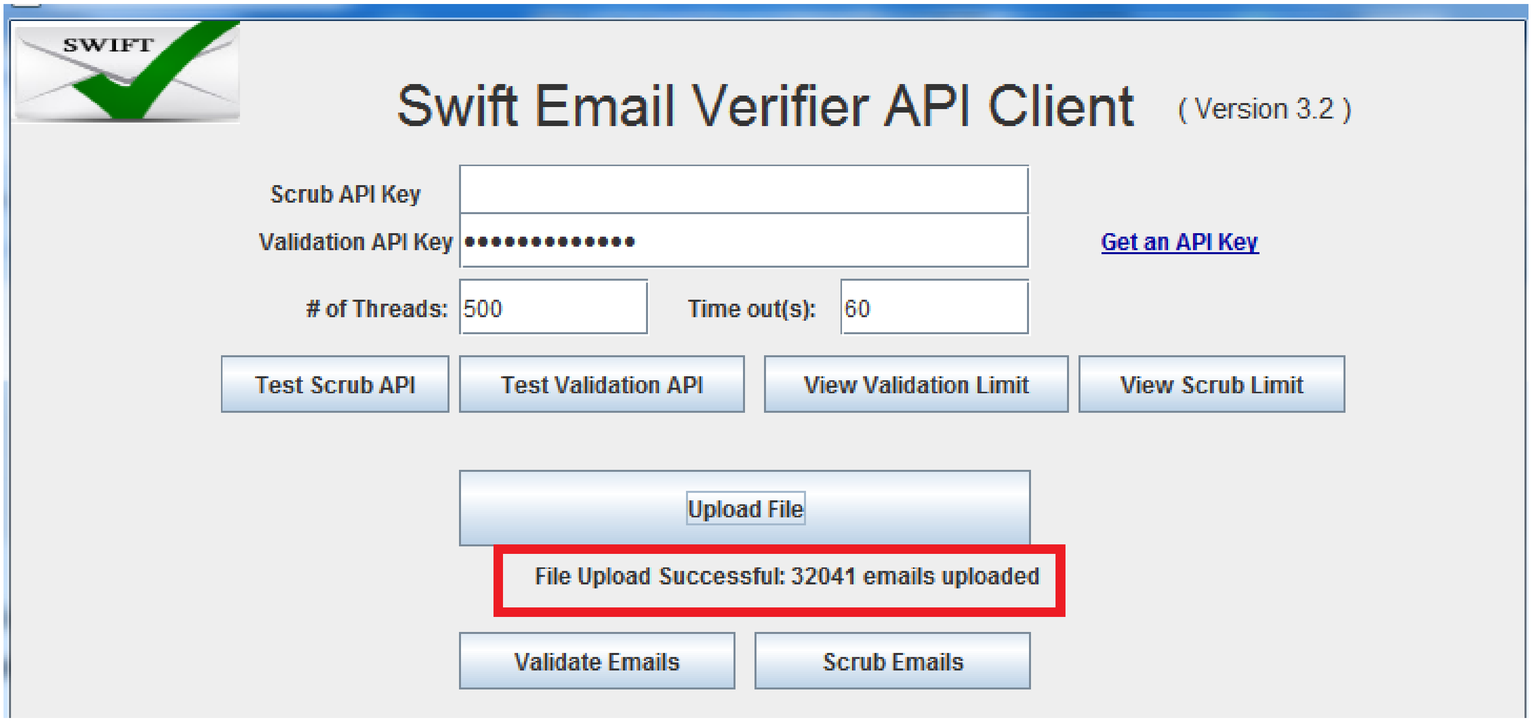
Time out(s): 60

Look In: Sample Email List Formats

File Name: format 1.txt

Files of Type: All Files

Depending on the size of the file you may see a progress monitor which displays the file upload activity followed by a ***File Upload Successful*** message and the number of emails that was successfully uploaded.



Scrub API Key

Validation API Key •••••••••••••

# of Threads: 500

Time out(s): 60

File Upload Successful: 32041 emails uploaded

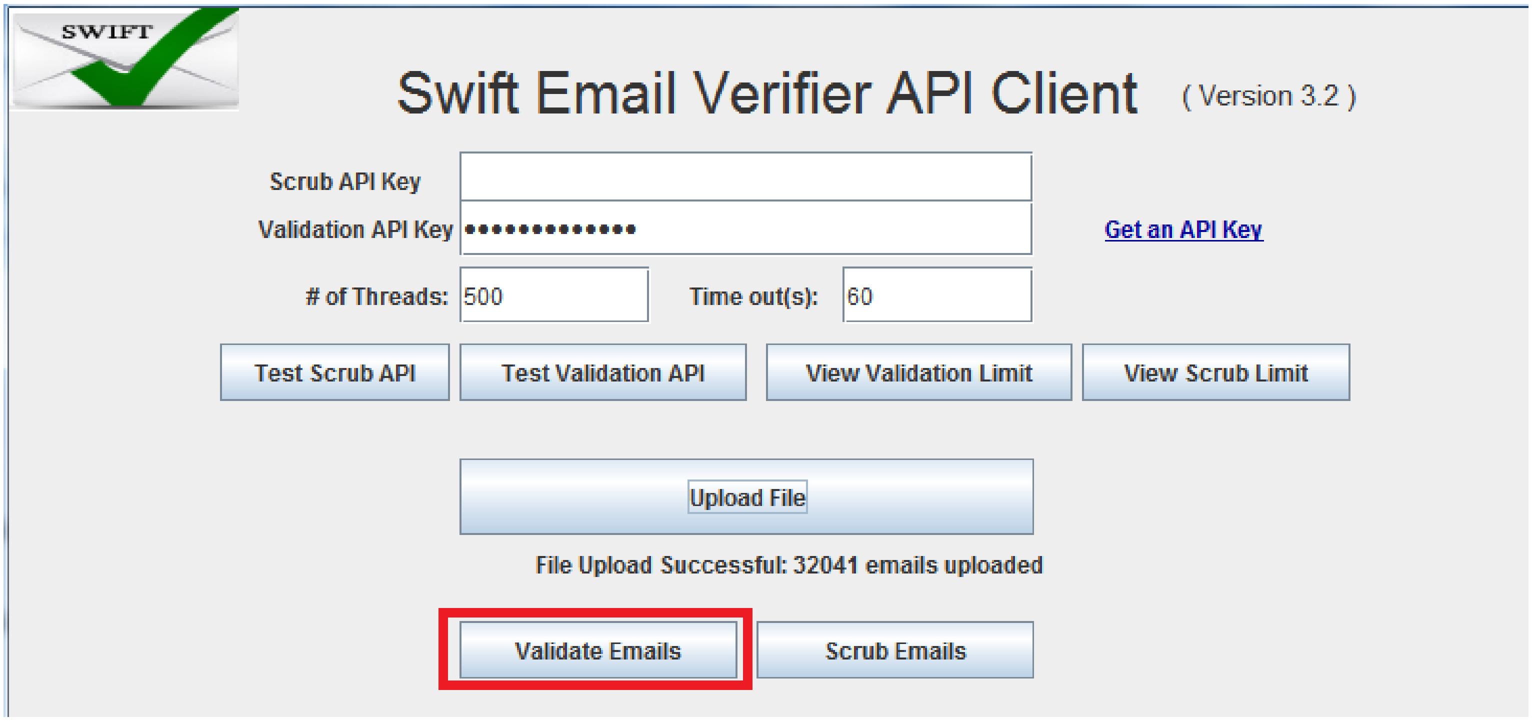
If the file is small, the application will directly show a ***File Upload Successful*** message.

A ***Validate Emails*** *and* ***Scrub and Emails*** buttons appears after the file is successfully uploaded.

Step2: Press the "Validate Emails" or "Scrub Emails" button to begin

If you have an email validation API key and you wish to validate your emails click on the "Validate Emails" button to start the validation process. If you have the email scrubbing API key and you are interested in using the email scrubbing function of the application, click on the "Scrub Emails" button .

Important Information: Please ensure that Microsoft Excel program is not opened or in use while the validation or scrubbing operation is in progress. Make sure you close all instances of Microsoft Excel that is running on your computer prior to running the program.



Swift Email Verifier API Client (Version 3.2)

Scrub API Key

Validation API Key ••••••••••••••

# of Threads: 500

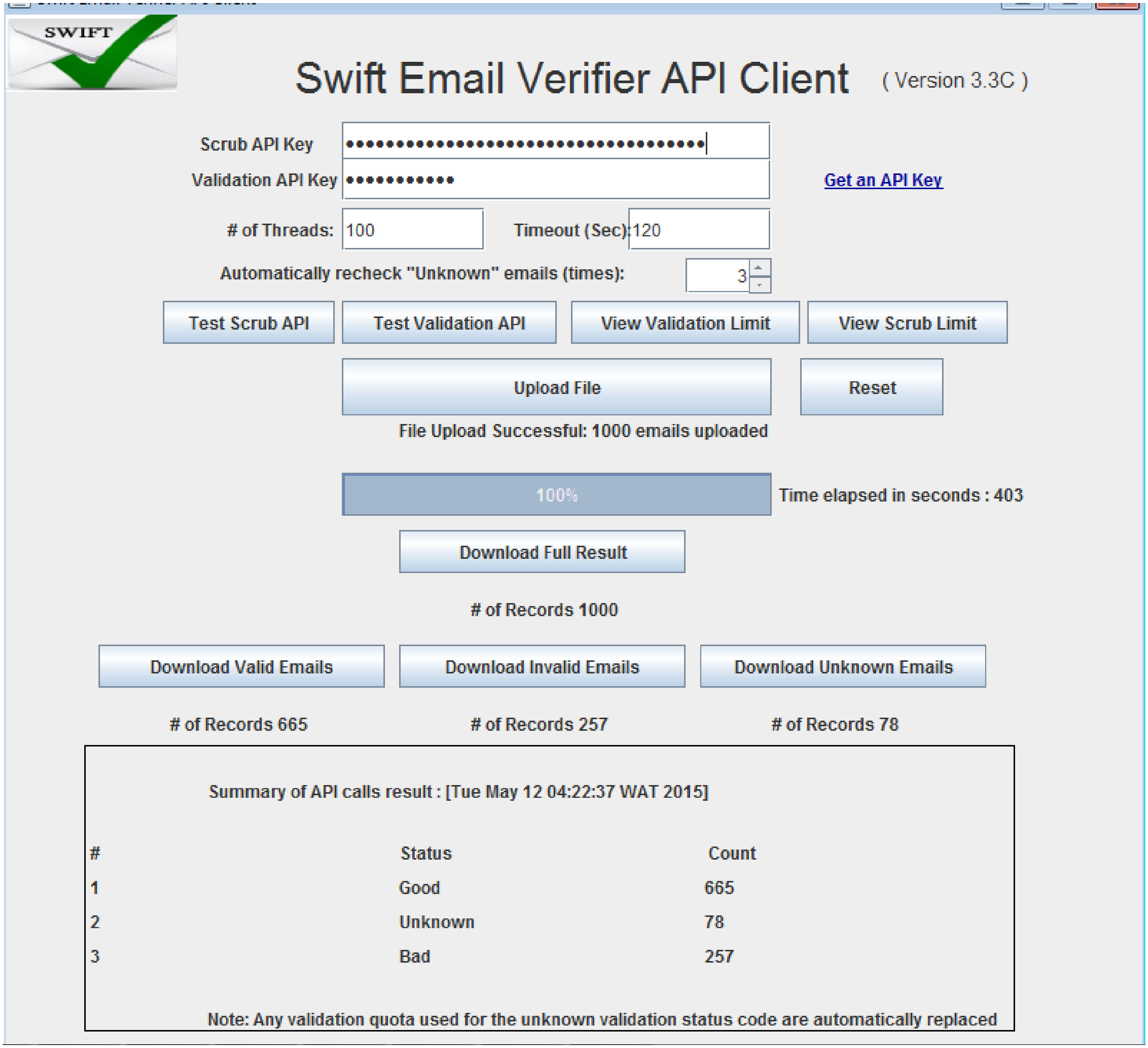
Time out(s): 60

File Upload Successful: 32041 emails uploaded

A progress bar will appear in a while and it will display the progress of the validation process.

After the progress is 100% complete, download buttons will appear to enable downloading of the results. The application is network failure tolerant and will automatically attempt to re-connect to start the validation process in case where your internet connection is lost. The program will retry to automatically re-connect up to 10 times before giving an “InternalError” status for the emails being verified during the network failure.

At the completion of the validation, 3 different result groups will be provided and the emails for each group can be downloaded as illustrated in the screenshot below.



Swift Email Verifier API Client (Version 3.3C)

Scrub API Key ••••••••••••••••••••••••••••••••••••

Validation API Key ••••••••••••

# of Threads: 100

Timeout (Sec): 120

Automatically recheck "Unknown" emails (times): 3

File Upload Successful: 1000 emails uploaded

100%

Time elapsed in seconds : 403

# of Records 1000

# of Records 665

# of Records 257

# of Records 78

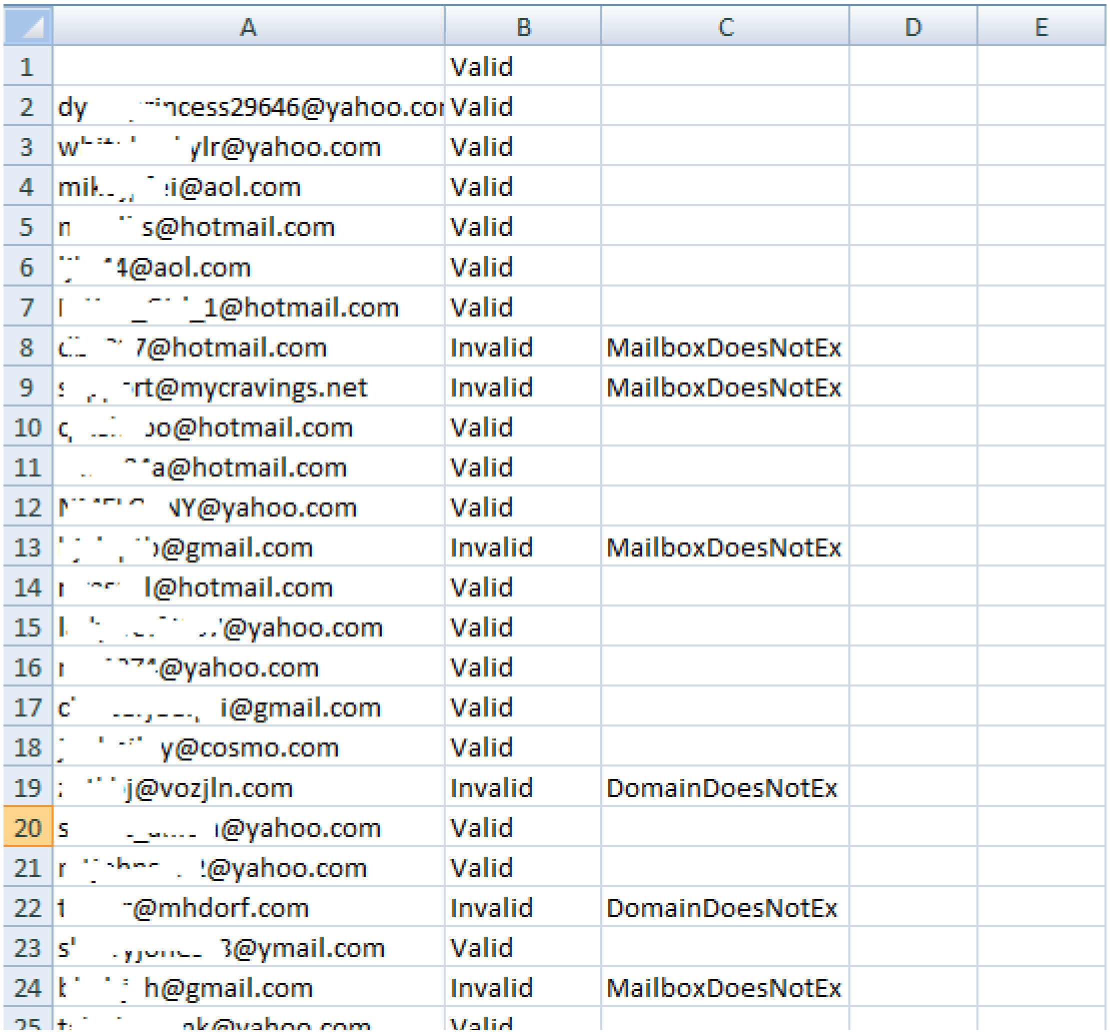
Summary of API calls result : [Tue May 12 04:22:37 WAT 2015]

#	Status	Count
1	Good	665
2	Unknown	78
3	Bad	257

Note: Any validation quota used for the unknown validation status code are automatically replaced

In addition, the “Full Result” can be downloaded which will contain all the results of the emails in a single file. A sample of the full result file in excel format is shown below:

Important Information: After the validation or scrubbing operation has completed and the results are now available, you must download the results files first before clicking on any other button on the program. Failure to do this might result to loss of your results!



	A	B	C	D	E
1		Valid			
2	dy[redacted]ncess29646@yahoo.coı	Valid			
3	w[redacted]ylr@yahoo.com	Valid			
4	mi[redacted]i@aol.com	Valid			
5	n[redacted]s@hotmail.com	Valid			
6	[redacted]4@aol.com	Valid			
7	[redacted]_1@hotmail.com	Valid			
8	c[redacted]7@hotmail.com	Invalid	MailboxDoesNotEx		
9	[redacted]rt@mycravings.net	Invalid	MailboxDoesNotEx		
10	c[redacted]o@hotmail.com	Valid			
11	[redacted]a@hotmail.com	Valid			
12	[redacted]Y@yahoo.com	Valid			
13	[redacted]@gmail.com	Invalid	MailboxDoesNotEx		
14	[redacted]l@hotmail.com	Valid			
15	l[redacted]@yahoo.com	Valid			
16	[redacted]@yahoo.com	Valid			
17	c[redacted]i@gmail.com	Valid			
18	[redacted]y@cosmo.com	Valid			
19	[redacted]j@vozjln.com	Invalid	DomainDoesNotEx		
20	s[redacted]@yahoo.com	Valid			
21	[redacted]@yahoo.com	Valid			
22	[redacted]@mhdorf.com	Invalid	DomainDoesNotEx		
23	s[redacted]3@ymail.com	Valid			
24	[redacted]h@gmail.com	Invalid	MailboxDoesNotEx		
25	t[redacted]nk@yahoo.com	Valid			

Scrubbing Emails:

In addition to the email validation capability of the JAVA verifier, the program also allows for scrubbing email addresses. The Email Scrubbing API is a real time email cleaning system that allows you to scrub email addresses against our millions of undesirable and bad email database such as bogus/stale email addresses, role accounts, disposable email addresses (DEA), publicly harvested/extracted email addresses and blacklisted emails/email domains.

By using our scrubbing API, you can detect and scrub out any known bad/bogus emails, stale emails, disposable emails, blacklisted emails and publicly extracted/harvested emails from your mailing lists which can damage your email marketing reputation or gets you blacklisted. By cleaning your email lists through our scrubbing service, you can avoid high email bounces which are key criteria in deliverability and sender reputation scoring.

Swift Email Verifier API Java Client- User Manual

The email scrubbing service is particularly useful to email marketers that have an old email lists or have recently acquired a list from a third party or list broker and wishes to verify or improve the quality of the lists before using them for email marketing campaigns. Purchased or rented lists could contain bad or stale and publicly harvested/extracted emails (potential honeypots/spamtraps). The outcome of using such lists without proper scrubbing could be a high number of bounces and abuse complaints.

Important Information!: What this email scrubbing service is not for

This email scrubbing service is not intended to help spammers detect Spamtraps or Honeypots setup by Anti SPAM organizations. Note that we do not assist in the detection of spam traps setup by ISPs and Anti-SPAM agencies.

In compliance with anti-spam laws and to avoid aiding and abetting spammers, we do not allow the use of our service to verify extracted or harvested email lists or for harvesting email addresses for SPAMMING purposes. The validations of rented or purchased mailing lists from reputable sources are permitted.

What is Checked by the Email Scrubbing API:

- ✓ **Bad/Bogus Email :** Bad or bogus email addresses can be detected
- ✓ **Spamtrap Indicators and Risky Email/Domains Check:** All known spamtrap indicators and risky email

  and email domains such as catch-all can be detected
- ✓ **Disposable Email Address Check:** This checks if the email is provided by a known Disposable Email Address (DEA) provider such as Mailinator, 10MinuteMail, GuerrillaMail and about 2000 more.
- ✓ **Role Accounts** such as admin@domain.com, webmaster@domain.com, support@domain.com etc
- ✓ **Known Blacklisted/Bogus emails and Email Domains Check:** All records matching our millions of known spammers emails, malicious or bogus emails and emails belonging to known spam domains emails database can be used to scrub your mailing lists and any matches are removed using our scrubbing API.. Please note that we do not assist in the detection of spam traps setup by ISPs and anti-SPAM agencies.

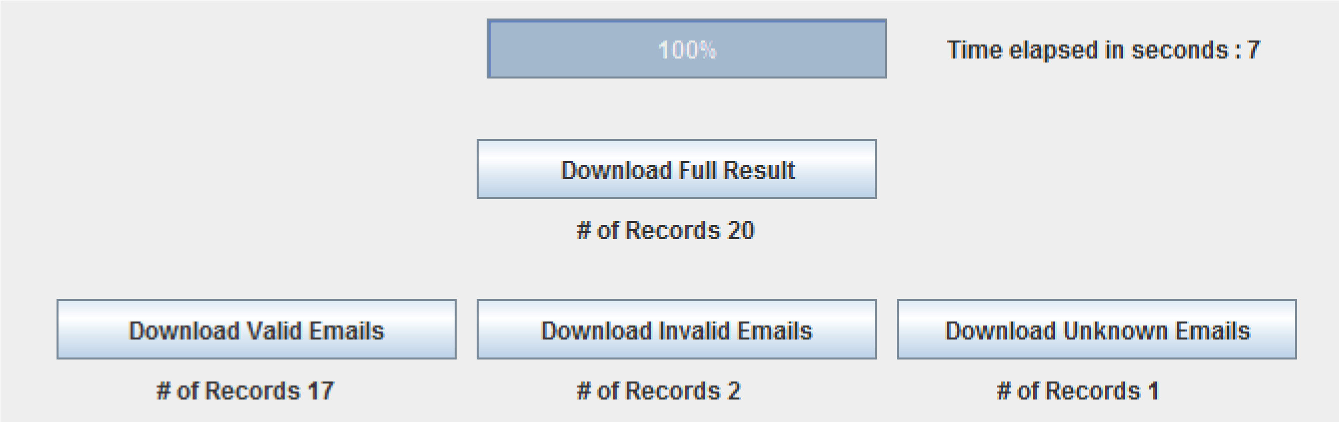
Note: Our scrubbing API helps to protect your list from damaging (could be deliverable) email addresses such as fake/bogus emails, emails, blacklisted emails and publicly extracted/harvested emails etc. However, it does not check if the email address actually exists in the mail server. If you need an email validation API that checks for actual mailbox existence, please visit our website to learn more. For a thorough email cleaning solution, we recommend the use of both APIs.

How the Scrubbing API Works:

When checking an email address against the scrubbing API, it is checked against our millions (Over 500 million) of email database of Role accounts, Disposable Email Address (DEA), Bogus/fake Emails and blacklisted, and bots emails. If the email address or email domain is found in the database, the email is marked “bad”. If no match is found, it is marked "Good".

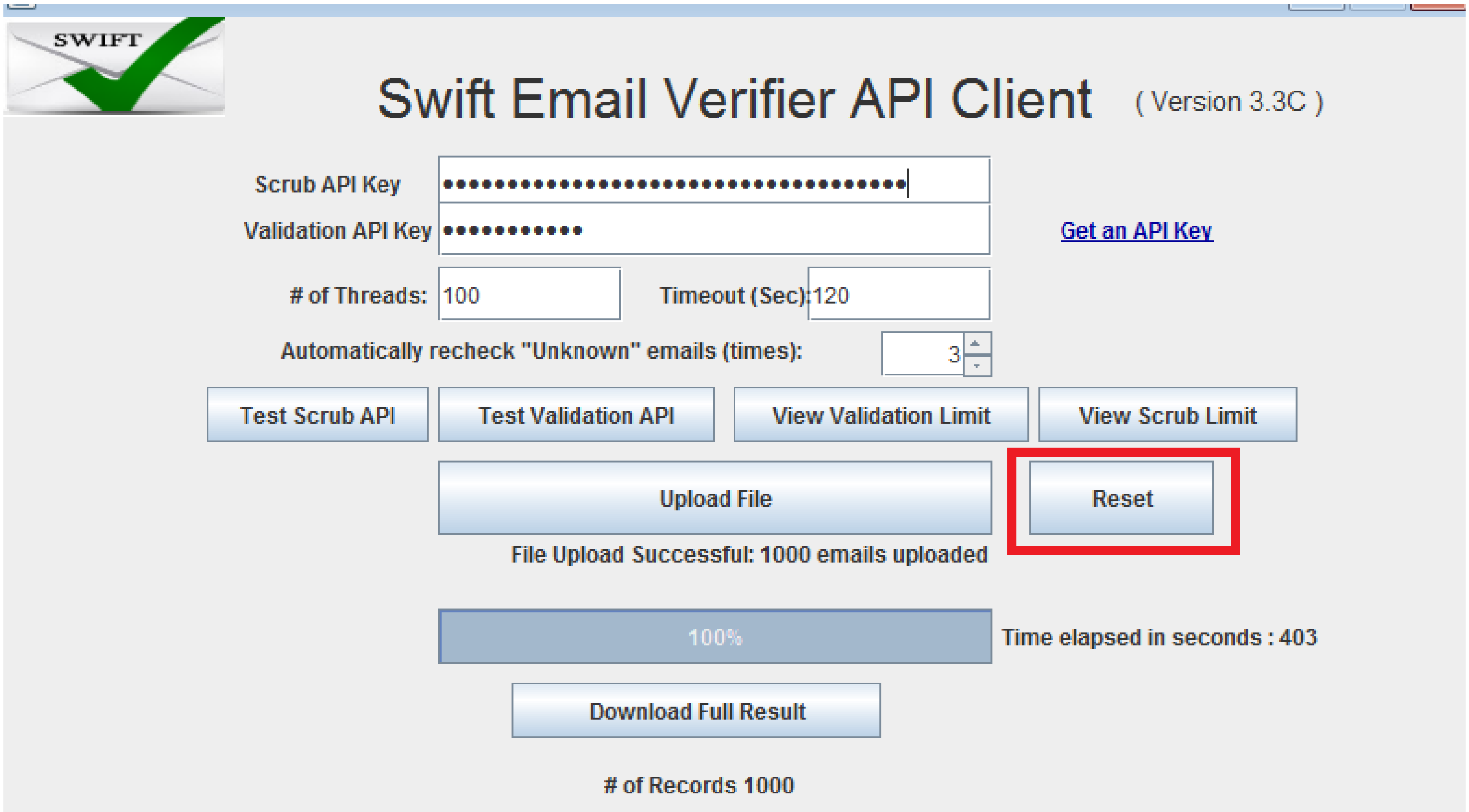
Please note that the email scrubbing API do not check if the emails actually exists on the mail server. If you need an email validation API that checks whether an email address actually exists on the mail server, please use our email validation API.

Click on the respective buttons to save the required group of emails. Once the validation is complete the download button appears on the program main interface. The results can be downloaded without an internet connection as well.



Resetting the Application

Each time you upload a list to the program, the files are saved in the program memory. The files are only deleted when you close the program or when you press the Reset button on the application. So each time you upload a file and you have downloaded the results, you need to click on the “Reset” button in case you wish to begin another validation immediately while the program is still open. This is required in order to flush all the emails it has saved in its memory.



Scrub API Key ••••••••••••••••••••••••••••••••••••

Validation API Key ••••••••••••

# of Threads: 100

Timeout (Sec):120

Automatically recheck "Unknown" emails (times): 3

File Upload Successful: 1000 emails uploaded

Time elapsed in seconds : 403

# of Records 1000

Unknown Results

The Unknown results are those emails which could not be verified due to one reason or the other. These unknown results in most cases results from Greylisting which is technology that reduces spam by rejecting initial email delivery attempts. The Greylisting works by returning a "Temporarily Unavailable" message to the sending mail server the first (and only the first) time a message is received from a given sender. Hence, it makes sense to retry these validations again after some time has elapsed.

We have compiled a list of all the current known issues which you may encounter while using our email validation system. You can download this document in the link below:

www.webemailverifier.com/issues.pdf

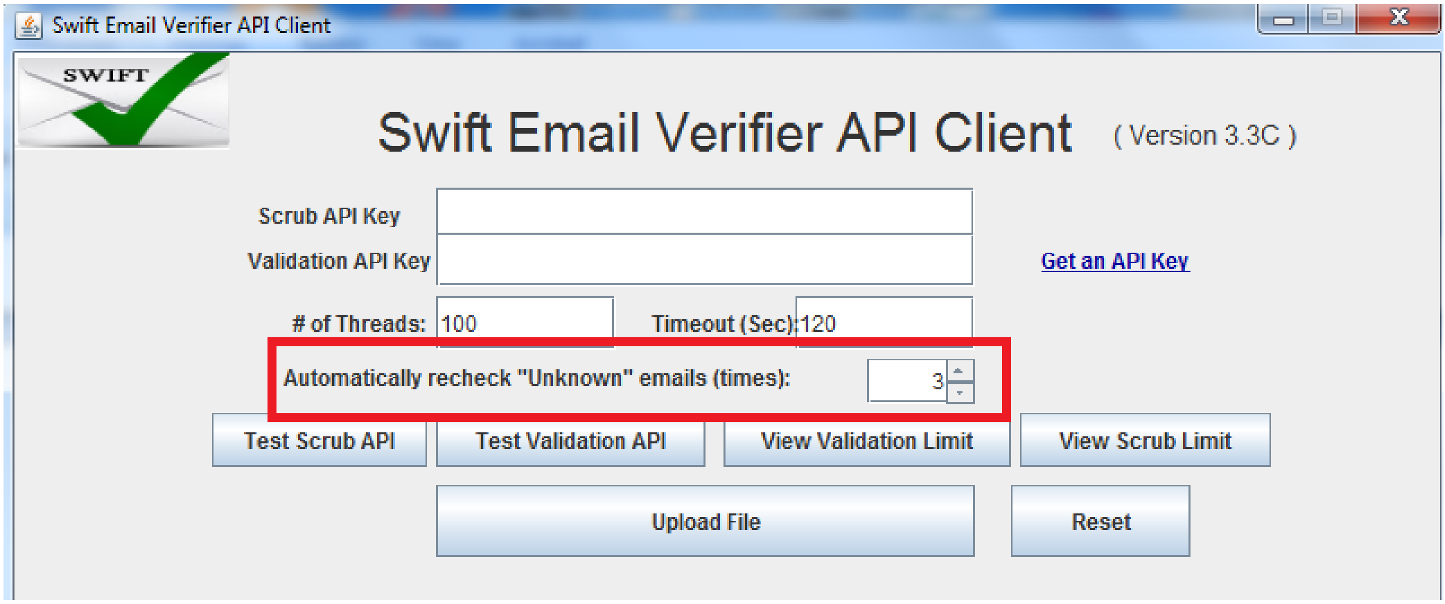
Also unknown results can also result from the inability to verify the emails by simulating a message sending to the recipient email server because the recipient email server requires that a REAL message is sent. Thus, it is impossible to verify whether the address is good or not. You won't know definitively until the message bounce because these mail servers won't cooperate or cannot be checked without sending a real message to them.

However, please be aware that some emails which return unknown results could be valid. Examples of such emails which are determined unknown by our API and which may be valid are:

- Disposable Email Addresses from email address providers, like Mailinator, 10MinuteMail, GuerrillaMail,etc
- Catch-all email addresses

- Temporarily Unavailable emails (Graylisting) and soft bounces

In order to minimize the number of unknown emails results returned by the program, the JAVA verifier uses an intelligent automatic multiple re-validation of unknown emails up to the number of times specified until a possible valid or invalid result is obtained. By doing this, the number of unknowns is greatly minimized.



Scrub API Key
Validation API Key
# of Threads: 100
Timeout (Sec): 120
Automatically recheck "Unknown" emails (times): 3

Automatic Replacement of API Credits Used for Unknown Results

With the new JAVA email verifier API client, all credits used for unknown results are now automatically replaced while performing validation jobs in real-time. All used credits for unknown emails will be automatically refunded to your API key quota without any action from you. Therefore you do not need to contact us to replace the credits anymore.

Email Validation Statuses and Status Codes

Our email validation API is a web service API and uses status codes to indicate API success or errors.

The email validation API defines the validity of an email address as follows using only 3 statuses:

Status	Description/Meaning
Valid	Mailbox exists and not handled by Catch-all domains or known to be a DEA
Invalid	Mailbox does not exists
Unknown	Mailbox could not be verified or is determined to be handled by a Catch-all domain, DEA, Greylisted,, SMTP/Mailbox timeouts, Temporary mailbox unavailability. Specific reason for failure is provided in the status codes.

Each of these Statuses is linked to the following status Codes:

Status Codes	**Meaning**
Mailbox Exists and Active	The email was successfully verified as Valid
Known Disposable Email Domain	This failure means that the email address is provided by a well-known disposable email address provider (DEA) such as mailinator.com
Syntax Error	This failure means that the email is not syntactically correct
Domain Does Not Exist	This means that the email domain has be found to be non-existent
Mailbox Not Found	This failure means that the mailbox for the provided email address does not exist.
DNS Query Error	This failure means that there was a DNS error when querying the MX server
SMTP Connection Blocked	This failure means that the external mail exchanger rejected the local sender address or the incoming connecting IP.
Mailbox Validation Error	This failure means that a timeout or error occurred while verifying the existence of the mailbox for the provided email address.
Mailbox temporary not reachable (Graylisting)	This failure means that the requested mailbox is temporarily unavailable; this is not an indicator that the mailbox actually exists or not but, often, a message sent by external mail exchangers with greylisting enabled.
Mailbox Not Reachable	This failure means that the email address could not be verified because the remote server was not responding

Catchall Email Domain	This failure means that the external mail exchanger under test accepts fake, non existent, email addresses; therefore the provided email address MAY be inexistent too. In most cases, these Catch-all domains are now setup by ISPs and ESPs as Catch-all Spam Trap domains specifically targeted to catch spammers using Dictionary Spam Attacks.
SMTP Connection Error	This failure means that a connection could not be established with the remote SMTP server
Curse Words Check	This status code indicates that the email address contained a curse word which most probably indicate it is a fake email address. E.g: fuck@yahoo.com
Fake Email Pattern Match	This status code indicates that the email address was detected to be fake using the API in-built fake email pattern detection algorithm. E.g: ususjsusjsjsjjss@yahoo.com
Typo Checking	This status code indicates that a typo error was detected for a known email domain such as : john@hotmaill.com
InvalidToken	An invalid API key was used. Please check the API key and make sure it is correct
NoMoreQueries	The allocated # of queries or requests for the API key has been exhausted.
InternalError	There was an unexpected error on our server.
InternalDBError	This error indicates that the API request failed due to database connection error from our server

Unable to get response from API:0	This error means that AEV dropped connection by timeout (set by user in the toolbar) when no data received from the API server. It may happen when user set low timeout and high number of Connections like 1000.
Invalid JSON Response	This error indicates that an error was received in the output of the results during the API call.

Email Scrubbing Statuses and Status Codes

Swift Email Verifier defines the reputation of an email address as follows using only 2 statuses:

Status	**Meaning**
Good	The reputation of the email is clean and no match was found in our database. This means that the email is not a DEA, Role account, not blacklisted nor bogus and does not match any email in our blacklist
Bad	The email is undesirable. The specific reason is listed in the status code.

Each of these Statuses is linked to the following status Codes:

Status Codes	**Meaning**
RoleAccount	The email is a role account which is an email address not associated with a specific person, but rather with an office, position, group or task such as admin@domain.com
DisposableEmailAddress	The email is a known disposable email address such as johndoe@mailinator.com

EmailBlacklist	Email is known to be blacklisted in all known Real-Time blacklists database
EmailDomainBlacklist	Email belongs to a known blacklisted domain
Bogus	Email is a known fake/malicious email or used by bots
InternalError	This error can occur when no response was received from our API server within the set Timeout. This can happen when using too high threads and low timeouts

Using Your Cleaned Emails in Third Party Email Delivery Services

Many companies and email marketers are now using third party email delivery services to handle the delivery of their transactional and marketing emails. By using a professional third party email delivery service, a higher inbox placement or delivery rate can be achieved. These third party email delivery services have relationships with various ISPs/ESPs and also provide strict acceptable “Bounce Rate” threshold for those who use their service to avoid being labeled as a server that delivers spam.

Although, the benefits of using a third party email delivery service is obvious, extreme care must be taken in order not to exceed the acceptable or permitted “Bounce Rate” for any email campaign you send through their platforms. Bounce rate is simply the percentage of emails that is returned undelivered when you send out your campaign. Although most third party email services do not explicitly specify their bounce rate limit, as a rule of thumb, anything between 10-15% may be considered high.

The Bounce Rate is expressed as a percentage and is calculated as follows:

$$Bounce\_Rate = \frac{No.\_of\_Bounces}{Total\_No.\_Emails\_Sent} x100$$

Undeliverable emails, email "bouncebacks" or "bounces" are becoming more and more of a challenge for email marketers these days. Hence, all third party email delivery services has a specific allowed or permitted bounce rate for every email campaign you send using their services. If you exceed this rate, your account may be suspended or deactivated. In most cases, these third party email delivery services are required by ISPs/ESP and Spam Advisory Groups such as Spamhaus to enforce the bounce rate thresholds and suspend any account that exceeds these thresholds in order to prevent Spam.

In order to avoid your third party email delivery service account suspension or deactivation, it is important that you review and adopt the following best practices before importing your cleaned emails into your third party email delivery service as follows:

1. After validating your list, save the VALID emails marked by the verifier. Do NOT add the emails marked as Unknown to the valid emails. As a rule, never upload the unknown emails to your third party email delivery service.

2. As we indicated above, never upload the emails marked as unknown by our email validation API to your third party email delivery service provider platform. Doing so may cause a lot of bounces and you may risk your account suspended.

To take care of unknown email results from our email verification API, you can use our new real-time bounce processing API application to verify the emails.

Frequently Asked Questions on Email Validation API

Question 1: How does your email validation API work?

Answer: Our email validation API is a simple and REST based API which can be used to validate emails effectively using the following order of validation processing:

- **Syntax Check:** This checks the email addresses and ensures that they conforms to IETF standards using a complete syntactical email validation engine
- **Mail Server Existence Check:** This checks the availability of the email address domain using DNS MX records
- **Mail Existence Check:** This checks if the email address really exists and can receive email via SMTP connections and sending email emulation techniques.
- **Catch-All Domain Email Check:** This checks if the email domain will receive all of the email messages addressed to that domain, even if their addresses do not exist in the mail server
- **Disposable Email Address Check:** This checks if the email is provided by a known Disposable Email Address (DEA) provider such as Mailinator, 10MinuteMail, GuerrillaMail and about 2000 more

Question 2: What do I need to start using your API in the JAVA program to validate emails?

Answer: First you must obtain the API key which allows you to authenticate to the API service. To obtain your API key, simply click go to https://www.bpsocks.com/member/signup.php. Once you have obtained your key, you can simply enter your key to activate the API. We offer a very flexible and affordable API pricing system. Our pricing plan is based on **$0.001** per email address validation.

Question 3: How is your email validations performed? Does it send out any email?

Answer: Email validations carried out through the API is done using 3 progressive levels automatically as follows:

- **Syntax** : This checks the email addresses and ensures that they conforms to IETF standards using a complete syntactical email validation engine
- **Email Server Existence** : This level checks the availability of the email address domain using DNS MX records
- **Mailbox Existence** : This is a deep level verification which attempts to check if the email address really exists and goes a step further to check if the email domain is a Catch-all domain (a domain that will receive all of the email messages

addressed to that domain, even if their addresses do not exist in the mail server). The Mailbox verification establishes SMTP dialogs with external SMTP servers and this level usually requires longer time depending on multiple network factors.

The API employs DNS and SMTP protocol functionalities to perform email address validations and absolutely avoids sending any email message to external mail servers.

Question 4: Is it possible to verify all emails with your email verifier API service? How does the system handle Unknown emails?

Answer: It is not possible to validate all emails due to multiple factors beyond our control. The Unknown results are those emails which could not be verified due to one reason or the other. These unknown results in most cases results from Greylisting which is technology that reduces spam by rejecting initial email delivery attempts. The Greylisting works by returning a "Temporarily Unavailable" message to the sending mail server the first (and only the first) time a message is received from a given sender. Hence, it makes sense to retry these validations again after some time has elapsed.

In addition, unknown results can also result from the inability to verify the emails by simulating a message sending to the recipient email server because the recipient email server requires that a REAL message is sent. Thus, it is impossible to verify whether the address is good or not. You won't know definitively until the message bounce because these mail servers won't cooperate or cannot be checked without sending a real message to them.

To take care of unknown email results from our email verification API, you can use our new managed hybrid email validation technology which uses the method of sending re-confirmation emails to validate emails with up to 98% success rate. For details on how it works, please contact us.

Question 5: Can I achieve low bounce rates with the email validation API?

Answer: One of the main reasons why you must validate your emails regularly is to ensure that you avoid high bounce rates when you send your campaign to your lists. When you send emails to invalid emails, the message will bounce. A bounced message is one that has been rejected by the recipient's email server. If your emails get bounce rates of over 10-15%, your email marketing service provider may likely disable your account until you can determine the cause of the bounces. This is because high bounce rates can get your email marketing service provider IPs blacklisted and will also negatively affect your sender reputation which will result to poor inbox deliverability. There are two types of bounces as follows:

- Hard bounces: These are bounces caused as a results of permanent failure during delivery (typically 5.x.x / Mailbox does not exist at the domain)

    Please see : http://www.basics.net/index.php/2011/07/27/e-mail-smtp-error-codes/

- Soft Bounces: These are bounces caused by temporarily failure such as Mailbox full errors ((beginning with a 4.x.x code as seen in above link)

With our email validation API, you will be able to verify your emails and detect a good number of emails that would have bounced (hard bounces) and these will be marked “Invalid”. Hence, you will be able to stay within the acceptable bounce rate limits typically permitted by email service providers. Emails with soft bounces will be marked “Unknown” and has be to revalidated. However, to identify emails with soft bounces which could turn out to become valid later, it is advisable to re-validate the unknown emails again after some days (1-3 days).

Question 6: Why are some invalid emails sometimes marked as Valid?

Answer: First, it is important to understand that our email validation technology uses the SMTP connection method to check whether a specific email address is valid or not by simulating email sending. However, due to certain multiple factors such as anti email harvesting technology, it is not possible to verify all emails with 100% success rate. This is because some mail servers such as public mail servers like Yahoo, AOL, etc have some measures in place which makes it impossible to accurately determine whether the email is valid or invalid because the mail servers will not cooperate and as a result the email address will be marked as valid when validated.

For example, Yahoo will always mark disabled or discontinued emails as Valid when verified. However, when you try to send to such disabled or discontinued emails, it will return this error message:

Remote server replied: 554 delivery error. Sorry your message to <email_address> cannot be delivered. This account has been disabled or discontinued.

For such mail servers, the only means to conclusively know if the email is valid or not is when the email bounce. Hence, it is recommended to use a suitable bounce handler to process the bounces for such non cooperating mail servers in order to obtain the invalid emails.

Question 7: How secure are my email addresses validated through your API servers?

Answer: We take your mailing lists confidentiality seriously. If using our API for email address validation via the verifier program, your email addresses are never stored on our servers. All checks are done in real-time. In addition, all API calls or requests are transmitted via Secure Socket Layer (SSL) technology to prevent any potential credential sniffing

Question 8: My question is not answered here. How can I get in touch with you?

Answer: Please contact us via our support center or email us at: service@bpsocks.com