



StreetSweeper®

API

Application Program Interface

PRODUCT GUIDE



Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced without the written permission of:

Pitney Bowes of Canada Ltd.
5500 Explorer Drive
Mississauga ON L4W 5C7
Canada

©2010-2019 Pitney Bowes of Canada Ltd. All rights reserved

Further Information:

Voice (905) 219-3000
Technical Support (905) 219-3120

Last Update: February 2019



Contents

Overview	4
Commands.....	5
Postal code and Optional Street Number Look-Up	6
OPTION 1 Address Correction/Validation	10
OPTION 2 Address Correction/Validation	11
Parsed Format.....	12
Near Matches.....	14
TCP Server Startup Options	18
Address Status Codes.....	18
Find Latitude and Longitude by Postal Code	17
NCOA (National Change of Address) Options	19
API TCP Functions using a Windows DLL.....	15



Overview

StreetSweeper software provides the ability to correct/verify addresses according to the standards required by Canada Post Corporation. Utilizing the TCP interface allows the user to connect to the StreetSweeper address correction engine from any PC connected either on a local network or over the internet. If you have the NCOA option installed you may also access it in a similar manner.

The TCP/IP interface is used by our customers in a variety of environments including web applications, data entry assist/verification and many others.

The following information gives the basics on how this API is implemented. Please contact us if you need additional information.



Commands

- From your application write the following data to the TCP port:

·SS_API_SEND:CMD=XXXX,CMD=XXXX,CMD=XXXX etc

Possible commands are:

EXIT	Terminate the API
MULTI_ON	Activates multiple street name look-up
MULTI_OFF	De-activates multiple street name look-up
UPPER	Results are returned in UPPER case
LOWER	Results are returned in UPPER and lower case
MIN_ON	API is minimized on the task bar
MIN_OFF	API panel is displayed
HIDE_ON	API panel is hidden
HIDE_ON_PERSIST	API panel is hidden and will be on restart until HIDE_OFF is sent
HIDE_OFF	API panel is restored
TASK_HANDLE:nnnn	Windows handle of task to restore focus to
TASK_NAME:aaaaaa	Windows task name (caption line), to restore focus to (Only one method of TASK should be used)
PARSE_ON	Returns parsed address information using API Option 2
PARSE_OFF	Turns off parsed address information when using API Option 2
CMD_ACK	Enables a response to the client when a CMD has been received (ACK)
CMD_ACK_OFF	Disable the acknowledgment to the client for a CMD command (This is the default)
RURAL_ON	Address information for rural postal codes will be recalled
RURAL_OFF	Only the city and province for rural postal codes will be recalled
TERM_ON	All responses will be terminated with a CR (hex \$0D\$)
TERM_OFF	Turns off the termination character (default)



API - Postal code and Optional Street Number Look-Up

From your application write the following data to the TCP port

- `SS_API_SEND:FIND_POSTAL_CODE=XIXIXI,YYYYYY` · where XIXIXI is the postal code and YYYYYY is the optional street number
- Start polling the TCP port, waiting for an answer

When a postal code has NOT been found the API will write the following back to the TCP client.:

· `SS_API_RESULT:NOT_FOUND`

When a postal code has been found the API will write the following data to the TCP client:

· `SS_API_FOUND:` followed by the following information

The information is contained in a 187 character record containing the API Signature, Number Found and one of the record types on the following pages. If there are multiple address locations for the postal code the first one is returned in the 187 character format. The second and subsequent ones are in a comma delimited format.

Civic # start, civic # end, street name, street type, street direction, spare,;

Note that each comma delimited record set end with a semi-colon (;). In the following example the postal code L3Y7V1 yields 5 records.

`SS_API_FOUND:` I,13 API Signature
Number found I4,3 Total matches

```
SS_API_FOUND: ··545·····RR·······LCD·1·······
·······NEWMARKET·······ONL3Y7V1·······
·······1339,2075,GREEN,LANE,E,;1590,
2056,GREEN,LANE,E,;18101,18685,LESLIE,ST,,;18106,18694,LESLIE,ST,,;
```

NOTE: The (·) character represents a space.



DATA	POSITION	COMMENTS
SS_API_FOUND:	1,13	API Signature
Number found	14,3	Total matches

Type One Record (Street Address)

Record type	17,1	Always 1
Suite number from	18,6	
Suite number to	24,6	
Civic number from	30,6	
Civic (street) number from suffix	36,1	
Civic (street) number to	37,6	
Civic (street) number to suffix	43,1	
Spare	44,1	
Street Name	45,30	
Street Type	75,6	
Street Direction	81,2	
City Name	83,30	
Province	113,2	
Postal Code	115,6	
LVR Type	121,2	
LVR Name	123,60	
Delivery mode type	183,1	Blank if LCP files are not installed
Delivery mode code	184,4	Blank if LCP files are not installed



Type Two Record (Street/Route Service)

Record type	17,1	Always 2
Route number	18,6	
Route type	24,6	RR or SS or MR
Civic number from	30,6	
Civic (street) number from suffix	36,1	
Civic (street) number to	37,6	
Civic (street) number to suffix	43,1	
Spare	44,1	
Street Name	45,30	
Street Type	75,6	
Street Direction	81,2	
City Name	83,30	
Province	113,2	
Postal Code	115,6	
LVR Type	121,2	
LVR Name	123,60	
Delivery mode type	183,1	Blank if LCP files are not installed
Delivery mode code	184,4	Blank if LCP files are not installed

Type Three Record (PO Box)

Record type	17,1	Always 3
PO Box number from	18,6	
PO Box number to	24,6	
Delivery Information (STN A etc)	30,30	
Blank	60,23	
City name	83,30	
Province	113,2	
Postal code	115,6	
LVR type	121,2	
LVR name	123,60	
Delivery mode type	183,1	Blank if LCP files are not installed
Delivery mode code	184,4	Blank if LCP files are not installed



Type Four Record (Route Service)

Record type 17,1 Always 4
Route number 18,6
Route type 24,6 RR or SS or MR
Delivery Information (STN A etc) 29,30
Blank 59,23
City name 83,30
Province 113,2
Postal code 115,6
LVR type 121,2
LVR name 123,60
Delivery mode type 183,1 Optional
Delivery mode code 184,4 Optional

Type Five Record (General Delivery)

Record type 17,1 Always 5
Delivery Information (STN A etc) 29,30
Blank 59,35
City name 83,30
Province 113,2
Postal code 115,6
LVR type 121,2
LVR name 123,60
Delivery mode type 183,1 Blank if LCP files are not installed
Delivery mode code 184,4 Blank if LCP files are not installed

Type Six Record (Rural Post Office Delivery)

Record type 17,1 Always 6
Blank 18,65
City name 83,30
Province 113,2
Postal code 115,6
Blank 121,67



API - OPTION 1 Address Correction/Validation

On-Site Installation

This function may be used to send a name and address to the API and have it validate/correct the information and optionally reformat it according to pre-determined options. The data is returned to the TCP client and is read back by the users application.

- From SETUP - API ADDRESS CORRECTION OPTIONS select the file format and options that you wish to use. This information is stored in the \StreetSweeper2\Auto_Ac\SAC_API.\$\$\$ file.
- Start the API and your application
- From your application write the following data to the TCP server: ·
SS_API_SEND:ADDRESS_CORRECT=your name and address data

The API will return the following information to the client that sent the message

SS_API_FOUND:your name and address data · The format of the data will be identical to the format your application sent



API - OPTION 2 Address Correction/Validation

Hosted Client Access or On-Site Installation

You may also send the data in a pipe (|) delimited format. Using this option is simpler and requires less data transmission overhead. The format for this is:

SS_API_SEND:FIND_CORR=Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province
| Postal Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 |

Normally you would send just the address 1, address 2, city, province, postal code information. You may want to include the Name and Misc information depending on the options selected in order to have the case changed or accented.

The data is returned in the same format:

SS_API_FOUND:Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province | Postal
Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 | City Abbrev. Name (13
characters)|;

PLUS parsed information when the Status Code = Valid or Corrected

Address Type|Street Number|Suffix|Unit Number|Street Name/Station Name|Street Type/Station
Type|Street Direction|PO Box Number|Service Type (RR/SS/GD)|Route Number|;

Note the semi-colon at the end of the data.

Parsed information will also be returned in the Misc 2 field. This is in a fixed field format described in detail on the next pages.

NOTE: This option can be enabled or disabled with the Command:

PARSE_ON or PARSE_OFF

The default is the parse option is OFF.



API - Parsed Format

The parsed output option will, in addition to returning the formatted address, return the parsed form of the address in USER FIELD TWO . In order to invoke this option it must be selected when setting the Address Correction options.

Addresses will be parsed for VALID and CORRECTED records but not INVALID records. If the address is INVALID we cannot guarantee the parsed output would be correct. Note that parsed output is returned in UPPER or UPPER and Lower case and with or without French accents, depending on the options that you have chosen. The street type will always be returned in the short form.

Record Types:	1 = Street Address	4 = Rural Route
	2 = Street Address (Route Service)	5 = General Delivery
	3 = PO Box	

TYPE 1	Street Address
RECORD TYPE	1 (1)
SUITE NUMBER	6
CIVIC NUMBER	6 (RIGHT JUSTIFIED)
STREET # SUFFIX	1 (BLANK=NONE, 1=1/4, 2=1/2, 3=3/4 OR A-Z)
STREET NAME	30
STREET TYPE	6
DIRECTION	2
FILLER	12

TYPE 2	Street Address (Route Service)
RECORD TYPE	1 (2)
SUITE NUMBER	6
CIVIC NUMBER	6 (RIGHT JUSTIFIED)
STREET # SUFFIX	1 (BLANK=NONE, 1=1/4, 2=1/2, 3=3/4 OR A-Z)
STREET NAME	30
STREET TYPE	6
DIRECTION	2
ROUTE KEYWORD	2 (RR or SS or MR)
ROUTE NUMBER	4
FILLER	6



API - Parsed Format

TYPE 3	PO Box
RECORD TYPE	1 (3)
BOX KEYWORD	7 (PO BOX or CP)
BOX NUMBER	6
AREA NAME	30
INSTALLATION TYPE	5
INSTALLATION DESC	15
TYPE 4	Rural Route (SS, RR, MR)
RECORD TYPE	1 (4)
ROUTE KEYWORD	7 (RR or SS or MR)
ROUTE NUMBER	6
AREA NAME	30
INSTALLATION TYPE	5
INSTALLATION DESC	15
TYPE 5	General Delivery (GD)
RECORD TYPE	1 (5)
GENERAL DEL	2 (GD)
KEYWORD	
FILLER	11
AREA NAME	30
INSTALLATION TYPE	5
INSTALLATION DESC	15



API - Near Matches

If the address is invalid, (the status code field starts with N), then an option is available that will return a number of near-matches for the address. This option is invoked by sending:

```
SS_API_SEND:SET_ADDRESS_RETURN_LIMIT=XX
```

This will return a confirmation of:

```
SS_API_FOUND:XX;
```

Where XX is the maximum number of items to return. Normally 20 is more than enough but 500 is the maximum setting.

```
SS_API_SEND:FIND_CORR=|||1 YONGE ST||TORONTO|ON||
```

In this case the address could not be verified/corrected so the closest matches were returned.

```
SS_API_FOUND:|||1 Yonge St||Toronto|ON||N-UR-S#-PC|||||;
```

```
SS_API_FOUND:5;
```

```
1,1,1,,,YONGE,ST,,TORONTO,ON,M5E1E5,A2,TORONTO STAR BLDG;  
1,1,1,,101,,YONGE,ST,,TORONTO,ON,M5E1Y5,B,CANADA POST ADMINISTRATION;  
1,1,1,,1801,,YONGE,ST,,TORONTO,ON,M5E1W7,B,TELSEC BUSINESS CENTRE;  
1,1,1,,2000,,YONGE,ST,,TORONTO,ON,M5E1N4,B,MCPHEE JENSEN/ PUNCH (DIRECT);  
1,1,1,,400,,YONGE,ST,,TORONTO,ON,M5E1E6,B,TORONTO STAR;
```

The format is:

Record types 1: (Street Address)

Record type, odd/even (1=odd,2=even),civic # start, Civic # end, suite # start, suite # end, street name, street type, street direction, city, province, postal code, LVR/Building type, LVR/Building name;

Record type 2: (Street served by route service)

Record type, odd/even (1=odd,2=even),civic # start, Civic # end, route keyword, route number, street name, street type, street direction, city, province, postal code, LVR/Building type, LVR/Building name;

Record type 3: (PO Box)

Record type, box number start, box number end, area name, installation type, installation description, city, province, postal code, LVR type, LVR name;



API - DLL Options

Using the SS-TCP.DLL

There are two files SS-TCP.DLL and SS-TCP.INI. The INI file contains the default IP address and PORT

number. You can modify this file as required.

E.g.

[Main]

IP_Address=127.0.0.1

Port=7500

NOTE: You must start the StreetSweeper TCP monitor before performing any DLL commands.

`S2-WIN32.EXE {space} SS-API_AC.PGM {space} -ARG {space} TCP:PORT`

Where PORT is the TCP port number

`void Start_Session(int port)`

where port - is a port number. Must match an open port for StreetSweeper

You have to call this function before you can send any requests. If the port number is 0 (zero) then the port number will be taken from the INI file. The IP address is ALWAYS taken from the INI file.

`void End_Session()`

You must close the session after you finish your requests.

You can use the DLL as a 'wrapper' for the TCP functionality that is documented on the TCP API pages. The syntax for this is described below:

There are a few functions available.

`Send_TCP_Request(char * m_request, char * reply, delay)`

`// where reply is char* reply, delay is int32`

Delay=Number of milliseconds to wait for a reply. In some cases the server may take longer to issue a reply. This tells the client to wait for a response.

.



Example:

```
Send_TCP_Request("SS_API_SEND:FIND_NCOA=|Name1|Name2|Name3|Address1  
|Address2|Postal Code|",reply, delay)
```

```
// where reply is char* reply, delay is int32
```

```
Send_NCOA_TCP_Request("SS_API_SEND:FIND_CORR:Name 1|Name 2|Name  
3|Address 1|Address 2|City|Province|Postal Code|Status Code|Misc  
1|Misc 2|Misc 3|Misc 4|Misc 5|Misc 6|",reply, delay)
```

```
// where reply is char* reply, delay is int32
```



API - Find Latitude and Longitude by Postal Code

- From your application write the following data to the TCP port:

·SS_API_SEND:FIND_LATLON=<POSTAL CODE>

Example:

SS_API_SEND:FIND_LATLON=K1A0B1

Return information:

"K1A0B1","45.37606","-75.69117"



API - TCP Server Startup Options

The startup syntax is:

S2-WIN32.EXE SS-API_AC.PGM -ARG TCP:XXXXXX

The TCP is followed by a colon and the XXXXX is the tcp address i.e. 10000



API - Address Status Codes

V = Valid	
C = Corrected	GK—General Deliver Keyword
N = Non-Valid	DN—Delivery Name
	DI—Delivery Information (i.e. STN MAIN)
LV—Large Volume Receiver	MN—Municipal Name
LR—Apartment Building	MA—Municipal Alternate
LB—Business Building	AB—Municipal Abbreviation
	PR—Province
C# - Street (Civic) Number	PC—Postal Code
SN—Street Name	IP—Invalid Punctuation
SA—Alternate Street Name	AL—Address information will not fit
ST—Street Type	ML—Municipal name will not fit
SD—Street Direction	RF—Valid address that has been reformatted
SK—Suite Keyword	
S# - Suite Number	Invalid addresses will have at least one of these codes
SM—Missing Suite in Apt or Business Bld.	RU—Rural Address
RK—Route Keyword	UR—Urban Address
BK—PO Box Keyword	US—USA Address
B# - PO Box Number	
RU-?? Valid rural postal code, address information not validated	
?? - Unknown Address	

StreetSweeper® API

Application Program Interface

PRODUCT GUIDE

Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced without the written permission of:

Pitney Bowes of Canada Ltd.
5500 Explorer Drive
Mississauga ON L4W 5C7
Canada

©2010-2019 Pitney Bowes of Canada Ltd. All rights reserved

Further Information:

Voice (905) 219-3000
Technical Support (905) 219-3120

Last Update: February 2019

Contents

Overview	4
Commands.....	5
Postal code and Optional Street Number Look-Up	6
OPTION 1 Address Correction/Validation	10
OPTION 2 Address Correction/Validation	11
Parsed Format.....	12
Near Matches.....	14
TCP Server Startup Options	18
Address Status Codes.....	18
Find Latitude and Longitude by Postal Code	17
NCOA (National Change of Address) Options	19
API TCP Functions using a Windows DLL.....	15

Overview

StreetSweeper software provides the ability to correct/verify addresses according to the standards required by Canada Post Corporation. Utilizing the TCP interface allows the user to connect to the StreetSweeper address correction engine from any PC connected either on a local network or over the internet. If you have the NCOA option installed you may also access it in a similar manner.

The TCP/IP interface is used by our customers in a variety of environments including web applications, data entry assist/verification and many others.

The following information gives the basics on how this API is implemented. Please contact us if you need additional information.

Commands

- From your application write the following data to the TCP port:

·SS_API_SEND:CMD=XXXX,CMD=XXXX,CMD=XXXX etc

Possible commands are:

EXIT	Terminate the API
MULTI_ON	Activates multiple street name look-up
MULTI_OFF	De-activates multiple street name look-up
UPPER	Results are returned in UPPER case
LOWER	Results are returned in UPPER and lower case
MIN_ON	API is minimized on the task bar
MIN_OFF	API panel is displayed
HIDE_ON	API panel is hidden
HIDE_ON_PERSIST	API panel is hidden and will be on restart until HIDE_OFF is sent
HIDE_OFF	API panel is restored
TASK_HANDLE:nnnn	Windows handle of task to restore focus to
TASK_NAME:aaaaaa	Windows task name (caption line), to restore focus to (Only one method of TASK should be used)
PARSE_ON	Returns parsed address information using API Option 2
PARSE_OFF	Turns off parsed address information when using API Option 2
CMD_ACK	Enables a response to the client when a CMD has been received (ACK)
CMD_ACK_OFF	Disable the acknowledgment to the client for a CMD command (This is the default)
RURAL_ON	Address information for rural postal codes will be recalled
RURAL_OFF	Only the city and province for rural postal codes will be recalled
TERM_ON	All responses will be terminated with a CR (hex \$0D\$)
TERM_OFF	Turns off the termination character (default)

API - Postal code and Optional Street Number Look-Up

From your application write the following data to the TCP port

- `SS_API_SEND:FIND_POSTAL_CODE=XIXIXI,YYYYYY` · where XIXIXI is the postal code and YYYYYY is the optional street number
- Start polling the TCP port, waiting for an answer

When a postal code has NOT been found the API will write the following back to the TCP client.:

· `SS_API_RESULT:NOT_FOUND`

When a postal code has been found the API will write the following data to the TCP client:

· `SS_API_FOUND:` followed by the following information

The information is contained in a 187 character record containing the API Signature, Number Found and one of the record types on the following pages. If there are multiple address locations for the postal code the first one is returned in the 187 character format. The second and subsequent ones are in a comma delimited format.

Civic # start, civic # end, street name, street type, street direction, spare,;

Note that each comma delimited record set end with a semi-colon (;). In the following example the postal code L3Y7V1 yields 5 records.

`SS_API_FOUND:` I,13 API Signature
Number found I4,3 Total matches

```
SS_API_FOUND: ··545·····RR·······LCD·1·······
·······NEWMARKET·······ONL3Y7V1·······
·······1339,2075,GREEN,LANE,E,;1590,
2056,GREEN,LANE,E,;18101,18685,LESLIE,ST,,;18106,18694,LESLIE,ST,,;
```

NOTE: The (·) character represents a space.

DATA	POSITION	COMMENTS
SS_API_FOUND:	1,13	API Signature
Number found	14,3	Total matches

Type One Record (Street Address)

Record type	17,1	Always 1
Suite number from	18,6	
Suite number to	24,6	
Civic number from	30,6	
Civic (street) number from suffix	36,1	
Civic (street) number to	37,6	
Civic (street) number to suffix	43,1	
Spare	44,1	
Street Name	45,30	
Street Type	75,6	
Street Direction	81,2	
City Name	83,30	
Province	113,2	
Postal Code	115,6	
LVR Type	121,2	
LVR Name	123,60	
Delivery mode type	183,1	Blank if LCP files are not installed
Delivery mode code	184,4	Blank if LCP files are not installed

Type Two Record (Street/Route Service)

Record type	17,1	Always 2
Route number	18,6	
Route type	24,6	RR or SS or MR
Civic number from	30,6	
Civic (street) number from suffix	36,1	
Civic (street) number to	37,6	
Civic (street) number to suffix	43,1	
Spare	44,1	
Street Name	45,30	
Street Type	75,6	
Street Direction	81,2	
City Name	83,30	
Province	113,2	
Postal Code	115,6	
LVR Type	121,2	
LVR Name	123,60	
Delivery mode type	183,1	Blank if LCP files are not installed
Delivery mode code	184,4	Blank if LCP files are not installed

Type Three Record (PO Box)

Record type	17,1	Always 3
PO Box number from	18,6	
PO Box number to	24,6	
Delivery Information (STN A etc)	30,30	
Blank	60,23	
City name	83,30	
Province	113,2	
Postal code	115,6	
LVR type	121,2	
LVR name	123,60	
Delivery mode type	183,1	Blank if LCP files are not installed
Delivery mode code	184,4	Blank if LCP files are not installed

Type Four Record (Route Service)

Record type 17,1 Always 4
 Route number 18,6
 Route type 24,6 RR or SS or MR
 Delivery Information (STN A etc) 29,30
 Blank 59,23
 City name 83,30
 Province 113,2
 Postal code 115,6
 LVR type 121,2
 LVR name 123,60
 Delivery mode type 183,1 Optional
 Delivery mode code 184,4 Optional

Type Five Record (General Delivery)

Record type 17,1 Always 5
 Delivery Information (STN A etc) 29,30
 Blank 59,35
 City name 83,30
 Province 113,2
 Postal code 115,6
 LVR type 121,2
 LVR name 123,60
 Delivery mode type 183,1 Blank if LCP files are not installed
 Delivery mode code 184,4 Blank if LCP files are not installed

Type Six Record (Rural Post Office Delivery)

Record type 17,1 Always 6
 Blank 18,65
 City name 83,30
 Province 113,2
 Postal code 115,6
 Blank 121,67

API - OPTION 1 Address Correction/Validation

On-Site Installation

This function may be used to send a name and address to the API and have it validate/correct the information and optionally reformat it according to pre-determined options. The data is returned to the TCP client and is read back by the users application.

- From SETUP - API ADDRESS CORRECTION OPTIONS select the file format and options that you wish to use. This information is stored in the \StreetSweeper2\Auto_Ac\SAC_API.\$\$\$ file.
- Start the API and your application
- From your application write the following data to the TCP server: ·
SS_API_SEND:ADDRESS_CORRECT=your name and address data

The API will return the following information to the client that sent the message

SS_API_FOUND:your name and address data · The format of the data will be identical to the format your application sent

API - OPTION 2 Address Correction/Validation

Hosted Client Access or On-Site Installation

You may also send the data in a pipe (|) delimited format. Using this option is simpler and requires less data transmission overhead. The format for this is:

```
SS_API_SEND:FIND_CORR=Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province
| Postal Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 |
```

Normally you would send just the address 1, address 2, city, province, postal code information. You may want to include the Name and Misc information depending on the options selected in order to have the case changed or accented.

The data is returned in the same format:

```
SS_API_FOUND:Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province | Postal
Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 |City Abbrev. Name (13
characters)|;
```

PLUS parsed information when the Status Code = Valid or Corrected

```
Address Type|Street Number|Suffix|Unit Number|Street Name/Station Name|Street Type/Station
Type|Street Direction|PO Box Number|Service Type (RR/SS/GD)|Route Number|;
```

Note the semi-colon at the end of the data.

Parsed information will also be returned in the Misc 2 field. This is in a fixed field format described in detail in the next pages.

NOTE: This option can be enabled or disabled with the Command:

PARSE_ON or PARSE_OFF

The default is the parse option is OFF.

API - Parsed Format

The parsed output option will, in addition to returning the formatted address, return the parsed form of the address in USER FIELD TWO . In order to invoke this option it must be selected when setting the Address Correction options.

Addresses will be parsed for VALID and CORRECTED records but not INVALID records. If the address is INVALID we cannot guarantee the parsed output would be correct. Note that parsed output is returned in UPPER or UPPER and Lower case and with or without French accents, depending on the options that you have chosen. The street type will always be returned in the short form.

Record Types:	1 = Street Address	4 = Rural Route
	2 = Street Address (Route Service)	5 = General Delivery
	3 = PO Box	

TYPE 1	Street Address
RECORD TYPE	1 (1)
SUITE NUMBER	6
CIVIC NUMBER	6 (RIGHT JUSTIFIED)
STREET # SUFFIX	1 (BLANK=NONE, 1=1/4, 2=1/2, 3=3/4 OR A-Z)
STREET NAME	30
STREET TYPE	6
DIRECTION	2
FILLER	12

TYPE 2	Street Address (Route Service)
RECORD TYPE	1 (2)
SUITE NUMBER	6
CIVIC NUMBER	6 (RIGHT JUSTIFIED)
STREET # SUFFIX	1 (BLANK=NONE, 1=1/4, 2=1/2, 3=3/4 OR A-Z)
STREET NAME	30
STREET TYPE	6
DIRECTION	2
ROUTE KEYWORD	2 (RR or SS or MR)
ROUTE NUMBER	4
FILLER	6

API - Parsed Format

TYPE 3	PO Box
RECORD TYPE	1 (3)
BOX KEYWORD	7 (PO BOX or CP)
BOX NUMBER	6
AREA NAME	30
INSTALLATION TYPE	5
INSTALLATION DESC	15
TYPE 4	Rural Route (SS, RR, MR)
RECORD TYPE	1 (4)
ROUTE KEYWORD	7 (RR or SS or MR)
ROUTE NUMBER	6
AREA NAME	30
INSTALLATION TYPE	5
INSTALLATION DESC	15
TYPE 5	General Delivery (GD)
RECORD TYPE	1 (5)
GENERAL DEL	2 (GD)
KEYWORD	
FILLER	11
AREA NAME	30
INSTALLATION TYPE	5
INSTALLATION DESC	15

API - Near Matches

If the address is invalid, (the status code field starts with N), then an option is available that will return a number of near-matches for the address. This option is invoked by sending:

```
SS_API_SEND:SET_ADDRESS_RETURN_LIMIT=XX
```

This will return a confirmation of:

```
SS_API_FOUND:XX;
```

Where XX is the maximum number of items to return. Normally 20 is more than enough but 500 is the maximum setting.

```
SS_API_SEND:FIND_CORR=|||1 YONGE ST||TORONTO|ON||
```

In this case the address could not be verified/corrected so the closest matches were returned.

```
SS_API_FOUND:|||1 Yonge St||Toronto|ON||N-UR-S#-PC|||||;
```

```
SS_API_FOUND:5;
```

```
1,1,1,,,YONGE,ST,,TORONTO,ON,M5E1E5,A2,TORONTO STAR BLDG;
1,1,1,,101,,YONGE,ST,,TORONTO,ON,M5E1Y5,B,CANADA POST ADMINISTRATION;
1,1,1,,1801,,YONGE,ST,,TORONTO,ON,M5E1W7,B,TELSEC BUSINESS CENTRE;
1,1,1,,2000,,YONGE,ST,,TORONTO,ON,M5E1N4,B,MCPHEE JENSEN/ PUNCH (DIRECT);
1,1,1,,400,,YONGE,ST,,TORONTO,ON,M5E1E6,B,TORONTO STAR;
```

The format is:

Record types 1: (Street Address)

Record type, odd/even (1=odd,2=even),civic # start, Civic # end, suite # start, suite # end, street name, street type, street direction, city, province, postal code, LVR/Building type, LVR/Building name;

Record type 2: (Street served by route service)

Record type, odd/even (1=odd,2=even),civic # start, Civic # end, route keyword, route number, street name, street type, street direction, city, province, postal code, LVR/Building type, LVR/Building name;

Record type 3: (PO Box)

Record type, box number start, box number end, area name, installation type, installation description, city, province, postal code, LVR type, LVR name;

API - DLL Options

Using the SS-TCP.DLL

There are two files SS-TCP.DLL and SS-TCP.INI. The INI file contains the default IP address and PORT

number. You can modify this file as required.

E.g.

[Main]

IP_Address=127.0.0.1

Port=7500

NOTE: You must start the StreetSweeper TCP monitor before performing any DLL commands.

`S2-WIN32.EXE {space} SS-API_AC.PGM {space} -ARG {space} TCP:PORT`

Where PORT is the TCP port number

`void Start_Session(int port)`

where port - is a port number. Must match an open port for StreetSweeper

You have to call this function before you can send any requests. If the port number is 0 (zero) then the port number will be taken from the INI file. The IP address is ALWAYS taken from the INI file.

`void End_Session()`

You must close the session after you finish your requests.

You can use the DLL as a 'wrapper' for the TCP functionality that is documented on the TCP API pages. The syntax for this is described below:

There are a few functions available.

`Send_TCP_Request(char * m_request, char * reply, delay)`

// where reply is char* reply, delay is int32

Delay=Number of milliseconds to wait for a reply. In some cases the server may take longer to issue a reply. This tells the client to wait for a response.

.

Example:

```
Send_TCP_Request("SS_API_SEND:FIND_NCOA=|Name1|Name2|Name3|Address1  
|Address2|Postal Code|",reply, delay)
```

```
// where reply is char* reply, delay is int32
```

```
Send_NCOA_TCP_Request("SS_API_SEND:FIND_CORR:Name 1|Name 2|Name  
3|Address 1|Address 2|City|Province|Postal Code|Status Code|Misc  
1|Misc 2|Misc 3|Misc 4|Misc 5|Misc 6|",reply, delay)
```

```
// where reply is char* reply, delay is int32
```

API - Find Latitude and Longitude by Postal Code

- From your application write the following data to the TCP port:

·SS_API_SEND:FIND_LATLON=<POSTAL CODE>

Example:

SS_API_SEND:FIND_LATLON=K1A0B1

Return information:

"K1A0B1","45.37606","-75.69117"

API - TCP Server Startup Options

The startup syntax is:

S2-WIN32.EXE SS-API_AC.PGM -ARG TCP:XXXXX

The TCP is followed by a colon and the XXXXX is the tcp address i.e. 10000

API - Address Status Codes

V = Valid	
C = Corrected	GK—General Deliver Keyword
N = Non-Valid	DN—Delivery Name
	DI—Delivery Information (i.e. STN MAIN)
LV—Large Volume Receiver	MN—Municipal Name
LR—Apartment Building	MA—Municipal Alternate
LB—Business Building	AB—Municipal Abbreviation
	PR—Province
C# - Street (Civic) Number	PC—Postal Code
SN—Street Name	IP—Invalid Punctuation
SA—Alternate Street Name	AL—Address information will not fit
ST—Street Type	ML—Municipal name will not fit
SD—Street Direction	RF—Valid address that has been reformatted
SK—Suite Keyword	
S# - Suite Number	Invalid addresses will have at least one of these codes
SM—Missing Suite in Apt or Business Bld.	RU—Rural Address
RK—Route Keyword	UR—Urban Address
BK—PO Box Keyword	US—USA Address
B# - PO Box Number	
RU-?? Valid rural postal code, address information not validated	
?? - Unknown Address	



StreetSweeper®

API version 4

Application Program Interface

PRODUCT GUIDE



Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced without the written permission of:

Pitney Bowes of Canada Ltd.
5500 Explorer Drive
Mississauga ON L4W 5C7
Canada

©2010-2022 Pitney Bowes of Canada Ltd. All rights reserved

Further Information:

Voice (905) 219-3000
Technical Support (905) 219-3120

Last Update: March 2022



Contents

Overview	4
Http/https Request Commands.....	5
Http/https Request JSON Format.....	6
Address Status Codes	8



Overview

StreetSweeper software provides the ability to correct/verify addresses according to the standards required by Canada Post Corporation. Utilizing the http interface allows the user to connect to the StreetSweeper address correction engine from any PC connected over the internet.

This REST API interface is used by our customers in a variety of environments including web applications, data entry assist/verification and many others.

The following information gives the basics on how this API is implemented. Please contact us if you need additional information.



Address Correction/Validation

Hosted Client Access

Send the data in a pipe (|) delimited format. Using this option is simple and requires less data transmission overhead. The format for this is:

```
http://xxx.xxx.xxx.xxx:port/v1.2/?SS_API_SEND:FIND_CORR=Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province | Postal Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 |~LIC=XXXX,CASE=U or L, FMT=Y or N
```

See the next page for options after the tilde (~).

Normally you would send just the address 1, address 2, city, province, postal code information. You may want to include the Name and Misc information depending on the options selected in order to have the case changed or accented.

The data is returned in the same format:

```
SS_API_FOUND:Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province | Postal Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 |City Abbrev. Name (13 characters)|;
```

PLUS parsed information when the Status Code = Valid or Corrected

```
Address Type|Street Number|Suffix|Unit Number|Street Name/Station Name|Street Type/Station Type|Street Direction|PO Box Number|Service Type (RR/SS/GD)|Route Number|;
```

Note the semi-colon at the end of the data.

NOTE: The Misc fields are not normally populated but may contain test that will be cased according to the option you are using.

Please see the next page for JSON format options.



Address Correction/Validation— json Option

Hosted Client Access

`https://dns_name:port/v1.2/json/?SS_API_SEND:FIND_CORR=Name 1 | Name 2 | Name 3 | Address 1 | Address 2 | City | Province | Postal Code | Status Code | Misc 1 | Misc 2 | Misc 3 | Misc 4 | Misc 5 | Misc 6 |~LIC=XXXX,CASE=U or L, FMT=Y or N`

The CASE= option is either U(pper) or L(ower) case and the FMT= option is either Y or N. This option will reformat invalid or valid addresses according to the CASE= option you are using.

When the /json/ option is selected the results are returned in a json formatted string.

```
JSON
{
  "ac_rec_typ": "1",
  "ac_nme_1": "",
  "ac_nme_2": "",
  "ac_nme_3": "",
  "ac_add_1": "13263 Germans Rd",
  "ac_add_2": "",
  "ac_city": "Peace River Regional District",
  "ac_region": "BC",
  "ac_postcode": "V1G 0C2",
  "ac_status": "-RF",
  "ac_status_code": "V",
  "ac_civic_num": "13263",
  "ac_civic_sfx": " ",
  "ac_unit": "",
  "ac_st_nme": "Germans",
  "ac_st_typ": "Rd",
  "ac_st_dir": "",
  "ac_bx_num": "",
  "ac_serv_typ": "",
  "ac_route_num": "",
  "ac_stn_typ": "",
  "ac_stn_nme": "",
  "ac_alt_mun": "Peace River",
  "ac_lat": "55.72323",
  "ac_lon": "-120.28599"
}
```



Address Correction/Validation— json Option

Hosted Client Access

The /json/ option returns the following values:

ac_rec_typ—1 or 2=street address, 3=PO Box, 4=Rural Route, 5=General Delivery

ac_nme_1, ac_nme_2, ac_nme_3 NOTE: Names are only included if you want the CASE changed

ac_add_1—address one

ac_add_2—address two

ac_city— city

ac_region—province

ac_postcode—postal code

ac_status—codes indicating changes (detailed on the next page)

ac_status_code—V=Valid, C=Corrected, N=Non-valid

PARSED FIELDS:

ac_civic_num—street number

ac_st_nme—street name

ac_st_typ—street type (ave, rd, st etc.)

ac_st_dir—street direction

ac_bx_num—PO box number

ac_serv_typ—RR, SS or GD route identifier or general delivery

ac_route_num—rural route number

ac_stn_typ—STN, LCD etc postal station identifier

ac_stn_nme—postal station name

ac_alt_mun—if the ac_city name is greater than 18 characters an alternate name is given

ac_lat—latitude

ac_lon—longitude



API - Address Status Codes

V = Valid	
C = Corrected	GK—General Deliver Keyword
N = Non-Valid	DN—Delivery Name
	DI—Delivery Information (i.e. STN MAIN)
LV—Large Volume Receiver	MN—Municipal Name
LR—Apartment Building	MA—Municipal Alternate
LB—Business Building	AB—Municipal Abbreviation
	PR—Province
C# - Street (Civic) Number	PC—Postal Code
SN—Street Name	IP—Invalid Punctuation
SA—Alternate Street Name	AL—Address information will not fit
ST—Street Type	ML—Municipal name will not fit
SD—Street Direction	RF—Valid address that has been reformatted
SK—Suite Keyword	
S# - Suite Number	Invalid addresses will have at least one of these codes
SM—Missing Suite in Apt or Business Bld.	RU—Rural Address
RK—Route Keyword	UR—Urban Address
BK—PO Box Keyword	US—USA Address
B# - PO Box Number	
RU-?? Valid rural postal code, address information not validated	
?? - Unknown Address	