VTP User Guide

Version V2.8

Canada Post Corporation
2701 Riverside Drive Suite E0245
Ottawa, Ontario

Copyright Notice

Table of Contents

Chapter 1 : epost Connect^{TM} Volume Transaction Processing Overview 1

Chapter 2 : Installation 3

System Requirements 3

Installing the VTP utility 3

Folder Layout 3

Required Configuration 4

Execution Samples 4

Chapter 3 : VTP Properties 5

Chapter 4 : Application Execution 6

Syntax 6

Chapter 5 : Using VTP Send 7

Conversation 7

Message 7

Assumptions 7

VtpSend example: 8

Command Line Options 8

Mandatory Keys 11

Backward Compatibility 12

Creating Conversations 12

Updating Conversations 12

Chapter 6 : Using VTP Track 14

VTPTrack example: 14

Command Line Options 14

Mandatory Keys 16

Illegal Keys 16

Chapter 7 : Using VTPReceive 17

VtpReceive example: 17

Command Line Options 17

Backward Compatibility..19

Chapter 8 : Status Log Files 20

VTPSend..20

VTPTrack 21

VTPReceive 22

Chapter 9 : Exception Files..23

VTP Send..23

VTP Track 23

VTPReceive 24

Chapter 10 : Error and Exit Codes 26

Error Codes 26

Exit Codes 28

Chapter 11 : Troubleshooting 29

Problems & Solutions..29

Chapter 12 : Limitations 30

Chapter 13 : Technical Support 33

Appendix..33

Usage Examples 33

Configuration Examples..34

Chapter 1 : epost Connect™ Volume Transaction Processing Overview

The Volume Transaction Processing (VTP) utility allows epost Connect™ customers (account holders) to create, distribute and track large volumes of epost Connect™ transactions (Messages) automatically without requiring the use of a browser. The intention is for customers to obtain the application and install it at their sites allowing them to work independently in the distribution and tracking of information to their own respective clients.

The VTP utility is composed of a set of VTP applications, provided as three separate executables:

1) The Volume Transaction Sending Application (VtpSend): provides the ability to automatically create conversations, create and send messages while preserving the core functionality of epost Connect™. This executable can either accept message data directly from the command line of supported operating system or an input file. The application reads data from command line or an input file and will create the message(s), connect to epost Connect™ Server and send multiple or single message(s). For each message, it will generate an entry in a log file showing either success or failure when creating the message(s). Any failure will also be logged in an exception file.

2) The Volume Transaction Tracking Application (VtpTrack): provides the ability to automate the creation of tracking data on messages sent by an account. It will accept selection criteria and generate tracking information for all messages meeting the criteria. The tracking information will be written to a text file.

3) The Volume Transaction Receiving Application (VtpReceive): provides the ability to download messages in bulk via the epost Connect™ server. It enables the ability to automatically download bulk messages from a Connect account either all at once or based on chosen selection criteria. To download all messages for an account, or to download a subset of messages based on a date parameter, in either case, all parameters are passed directly to the executable.

The VTP utility applications are available to qualified epost Connect™ customers through a request to their Canada Post sales representative. An account token will be created and distributed with the VTP utility to the customer in a manner mutually agreed upon at the time of sale.

The application package will be delivered as an MS Windows based zipped file which can be easily extracted by the recipients directly to any desired location on their platform. Once installed and configured, customers are able to create, distribute, receive and track message packages which utilize the epost Connect™ secure server for delivery. Each transaction will be executed in a totally secure manner employing an SSL connection with the epost Connect™ server. Customers can rest assured that all of their information is stored within Canada's borders and safe from the reach of unauthorized parties and like regular mail, are protected under the Canada Post Corporation Act and the Criminal Code of Canada.

Chapter 2 : Installation

System Requirements

The following is required to install and run the VTP utility:

- Minimum Memory (RAM): 128 MB
- Minimum Disk Space: 1 GB
- Unzip utility: Windows Winzip, TAR or similar compatible with ZIP compression

A valid epost Connect™ account and associated token is required to run the applications, and can be obtained from your Canada Post sales representative. The account token is valid for the owner's usage only.

Installing the VTP utility

The VTP utility will be delivered as an *epost.connect.zip* file. This file contains the VtpSend and the VtpReceive applications that are associated with the VTP utility. Upon reception, the user must extract the epost.connect zip file to the desired location on his/her server/PC.

The extracted files will be placed under one folder named *epost.connect* which contains a sub-folder named *vtp*. The *vtp* sub-folder contains all other important sub-folders, named *Layout* as indicated below.

Folder Layout

The extracted VTP utility has following folder structure:

Note that **$HOMELOC** = the location in which the epost.connect.zip file was extracted.

$HOMELOC/epost.connect/vtp – contains the following:

- **bin** – contains the VtpSend, VtpTrack, and the VtpReceive application macros
- **conf** – contains a configuration file indicating the epost Connect™ server to be used
- **lib** – contains libraries required by the VTP applications
- **logs** – default location for log files. Log folder can be changed in configuration or by executable (see –i and -e keys)
- **jvm** – all files related to the Java virtual machine required by the application
- **doc** – contains a version of the VTP user guide (this document)
- **tdir** – default location for output of VtpReceive
- **samples** – a directory containing 6 examples of .cmd files that illustrate various features of VtpSend and VtpReceive. The files contain operational samples that may be used with small

modification or as a guide to create operational scripts. Each file contains comprehensive documentation on how to use VtpSend and VtpReceive.

Required Configuration

The VTP utility now requires the following configuration steps:

- Edit the "$HOMELOC/epost.connect/vtp/samples/setenv_new.bat " file by right-clicking on the file and selecting **Edit** from the submenu.
    - The file will contain a single line, similar to the following:

      set VTP_HOME=D:/Nov20/epost.connect/vtp

    - Change "D:/Nov20" to the location where the contents of the epost.connect.zip file were extracted. For example, if the content was extracted directly to "C:/", change "D:Nov20" to "C:/" in the file. The line will now look like:

      set VTP_HOME=C:/epost.connect/vtp

    - Save and close the file

Execution Samples

Useable examples of the VtpSend, VtpTrack and VtpReceive applications are provided in the *$HOMELOC/epost.connect/vtp/samples* folder. These samples (provided as **.cmd** files) illustrate a variety of usage options for all three applications and require some custom manual changes prior to being executed. It is recommended that upon installation, the user goes to the *$HOMELOC/epost.connect/vtp/samples* folder, views and edits the files and then executes them. The **.cmd** examples outline the functionality and edits required for each file. Use the remainder of this document to understand the content of the file and intended behaviour. After the file has been edited correctly, simply double-click on the desired **.cmd** file to observe the behaviour of the scenario. These examples may be modified by the user to fit his/her desired commercial behaviour. Alternatively, the user can directly create his/her own scenarios.

Chapter 3 : VTP Properties

The VTP utility must identify a corresponding epost Connect™ server in which to communicate and host the service. This server information is identified in a file named *vtp.properties*, located in the *$HOMELOC/epost.connect/vtp/conf* directory._By default, the server being identified is that of Canada Post. Therefore, it is not advisable to change the data contained in this file unless instructed to do so by Canada Post personnel.

The following property can be set using this file:

Property	Default Value	Description
ServerName		URL of epost Connect™ server available for external users of service.

Chapter 4 : Application Execution

VTP receives processing instructions from command line parameters. Command line can be specified as one line following the executable or as multiple lines from an input file. The number of parameters in the command line is only limited by the particular operating system running VTP.

Syntax

Command line format: <executable> <mandatory parameters> [optional parameters]

Command Line Component	**Description**
<executable>	Command line always starts with an executable: VtpSend, VtpReceive or VtpTrack
<mandatory parameters>	Each executable has mandatory parameters. Executable fails if one of mandatory parameters is not provided. For example: parameter identifying VTP access token is mandatory for each executable, parameters identifying conversations are mandatory for 'vtpSend'.
[optional parameters]	Each executable has optional parameters. Executable specifies default behavior if one of optional parameters is not provided. For example: parameter identifying language of communication is optional, VTP assumes English if it is missing.

Command line parameter format: <key> [optional value] [optional sub-parameters]

Parameter Component	**Description**
<key>	Keys are specific to each executable. Keys identify entities in the command line. Keys can appear in any order. Keys are prefixed with a hyphen (**-**). For example: key **-s** identifies conversation subject, key **-m** identifies message file.
[optional value]	Keys may have a value. Most of the keys expect a value as one or more arguments. Some keys are Boolean switches: value is set to true if key is presented or to false if key is missing. For example: key **-s** must have value of conversation subject

Chapter 5 : Using VTPSend

The VtpSend application (VtpSend) is used to create and send messages via the epost Connect™ server. The VTPSend application provides the ability to automatically create conversations (groups of messages with a single subject line or title) and create/send single or multiple batch style messages. To send a single message, all required parameters are passed directly to the executable. To send multiple messages, an input file containing specifications for each message is created and passed to the executable. The *epost.connect\vtp\samples* directory contains examples of **.cmd** files that may be utilized to familiarize oneself with these different execution styles.

Conversation

A Conversation is a secure grouping of subject related messages by which the conversation owner communicates to recipients.

- Messages are organized in Conversations
- Each Conversation has only one specific subject line
- A Conversation will include only messages related to the provided subject

Message

A message is a secure communication that the conversation owner sends to you. It may also contain attachments.

- The message envelope arrives closed and must be opened by the intended recipient.
- The openned message consists of text and/or file attachments.
- Messages have an expiry date set by the sender; if the message has expired then you must contact the sender to resend the message in order to view it.
- To retain the content of the message, you must copy the message text and download any attached file to your local storage device.

Assumptions

1) It is assumed that all attachments exist as files and are accessible to the VtpSend application
2) Text for the message, if any, has been written to an ASCII file that is accessible to the VtpSend application

VtpSend example:

VtpSend executes in command line style with key-value pairs that follow the executable call. The following example illustrates a command line sending a message to a single recipient containing a message body and attachment.

%VTP_HOME%/bin/VtpSend -a test-app1 -s "Message with attachment" –to -lang en -m Files/message.txt –att Files/Zombies.pdf

Note that it is practical to issue these commands from inside the script files (**.cmd** files for DOS type operating systems) which themselves can be executed by double-clicking the file instead of typing a string of commands directly on the command line.

The result produces the following key-value pairs:

Key	Value	Description
a	test-app1	The sender's epost Connect™ access token for VTP
s	"Message with attachment"	Message subject/conversation title
to		The receiver's email address
m	Files/message.txt	The relative location of the file containing the message body text (note that this file path is relative to the location of execution)
att	Files/Zombies.pdf	File to be attached to the message

Command Line Options

The following table explains all command line parameters, where support options are: n/s – not supported (retired), empty – for supported keys:

Key	Command	Default Value	Description
-a	Account identity	Not Applicable	The account access token associated with this instance of VTP. **Note**: Previous versions of VTP required the username and password of an account. This has now been deprecated to a single access token. Therefore, pre-existing VTP setups must request a single access token for their account in order to use this VTP package. Example 1: -a test-app1 Example 2: -a 730d75db4a41434bb60a32
-to	List of participants	Not Applicable	Email addresses of the participants. For multiple email addresses, the list should be separated by commas and/or spaces. Example1: -to user1@email.com user2@email.com

			Example2: -to user1@email.com, user2@email.com
-toe	List of participants	Not Applicable	This key provides privilege to the participants to reply to the message via UI. This key can be used in conjunction with the **-to** key Example1: -toe user1@email.com user2@email.com Example2: -toe user1@email.com -to user2@email.com (In the 2nd example, user1 will be able to reply to the message via UI and user2 can only view the message via UI)
-s	Subject	Not Applicable	Conversation/Message subject. This will appear on the header as the subject line of the email notification. This also identifies conversations when using backward compatible keys. The text can be enclosed in double quotes (" "). Double quotes cannot be placed in the middle of the text. Special characters can be included but have to be preceded by the Escape (Esc) key required by the command shell. Example1: -s Policy #123456 Example2: -s “This subject line has quotations”
-m	Message text	Empty	The full/relative path to the file containing the text message body. This key may be omitted only if there is an attachment file to send. Connect will create a message with an empty body if this key is omitted. Example: -m D:/folder1/folder2/message.txt
-att	Attachments	None	Full/relative path to attachment file. It can be single file or multiple files separated by comma and/or spaces. Full path containing commas or spaces has to be in double quotes (" "). Example 1: -att D:/folder1/folder2/my-file.pdf Example 2: -att “folder1,2/my file.pdf”
-sec	Security	None	Any argument of this key is the authentication answer for all participants being connected to the referenced conversation. Participants will be required to enter this value (authentication answer) to access the conversation on the server. The value must be in double quotes (" ") if it contains spaces. Example: -sec “fluffy bunny”
-lang	Notification Language	en	Language for invitation/notification emails.

			English: **e** or **en** French: **f** or **fr** Note that if this key is not specified on the command line, English (en) is assumed. Example: -lang fr
-i	Include message into notification	None	This key is re-used for including a personal message in the invitation email for all receivers. When this key is used on its own, the text of message file specified with the **-m** key will be included in the notification email. This is backward compatible usage of **-i** . Example: -i When this key is used with argument, the text of personal messages is taken from the file specified by the full path. Connect includes personal messages to all participants every time this key is provided. **Note**: It is recommended that for those using the new VTP to specify a file in the **-i** command (as shown below) due to possible exposure of secure content located in the message file. When **http:** is included in the personal message, a hyperlink will be created. The maximum size of the personal message contained within the notification server is 500 characters. Any remaining characters will be truncated. Example: -i D:/folder1/folder2/invitation.txt
-exp	Message expiration	10 days	This is the key used to indicate the expiration day of the message, and is in the unit of **days**.
-f	Multiple commands	False	This is used for sending multiple conversations using a command line call. The file should be a text file containing command keys. The file will be read and processed from top to bottom. A working example is provided in the *epost.connect/vtp/samples* directory. Example: -f input-file.txt Please see use of the **-f** key by viewing the *epost.connect/vtp/samples* directory.
-l	Output log	Logs are written to	The status of all transactions (successful or failure)

	file	: ‘epost.connect/vtp /logs’ and have default name: ‘*output-<date-time>.log*’	are written to the log file. By default, it will generate a log file named *output-<date-time>.log* where the "date-time" is the current date and time of the user’s PC terminal. This key allows the user to output the status log file to a directory other than the default of *epost.connect/vtp/logs* and have a name other than the default above. The format of log details are as follows: Package ID, Timestamp, Subject, Sender's account name, Recipient's email, Package status Example: -l D:/folder1/folder2/vtp.log Where: • **D:/folder1/folder2** is the specified destination of the log file, and **vtp.log** is the desired log file name
-e	Error log file	Exception logs are written to : ‘epost.connect/vtp /logs’ and have default name: ‘*exception-<date-time>.log*’	All exceptions are written to an exception log file. By default, VTP will generate an exception log file named *exception-<date-time>.log* in the *epost.connect/vtp/logs* directory where the 'date-time' is the current date and time of the user’s PC terminal. This key allows the user to select an alternative directory and name for the exception file. The format of exception log contents are as follows: Timestamp, Subject, Sender's account name, Recipients email address, Error code, Reason of failure. If there is no exception thrown, the exception log will still be present and will be empty (file size will be 0). Example: -e D:/folder1/folder2/vtp-error.log Where: • **D:/folder1/folder2** is the specified destination of the exception log file, and **vtp-error.log** is the desired exception log file name.
-help	Help		Provides help information about command line.

Mandatory Keys

VtpSend has the following mandatory key groups; one of the following combinations must be used for a valid execution:

Key Group	Keys	Description
Creating conversation or updating existing	-a –s [-m and/or –att]	Connect will create new conversation with no participants if subject doesn’t match any existing

conversations by subject and participants	<the rest of command line>	conversation. Example: -a ABC123 -s Policy #12345 …
Reading input from command file	-a -f	Command file contains rest of command line. The file associated with the **-f** key must contain subject line [-s] , a message [-m] and/or an attachment [-att] Example: -a ABC123 -f D:/folder1/folder2/input.txt

Backward Compatibility

VtpSend can create new conversations and post messages into existing conversation using existing **PosteCS** keys. The existing scripts have to refer only to the new executable.

Backward Compatibility allows migrated PosteCS customers to continue using the same interfaces and processes for operations on Connect.

VtpSend provides default behavior for each executable matching current PosteCS experience as close as possible. However, not all epost Connect™ features can be accessed using the existing commands. New epost Connect™ clients are encouraged to use new command keys which allow full access to all epost Connect™ operations and making VtpSend processing faster and more reliable.

Creating Conversations

By default, epost Connect™ creates a new conversation each time VTP submits a unique combination of subject and participant emails. For example, (irrelevant keys are omitted):

-s Policy #12345 -to user1@email.com user2@email.com -m message.txt -att attachment.pdf

The processing is as follows:

1. Search the Owner's data-set of conversations whose subject is "Policy #12345", and shared with the user1@email.com and user2@email.com email addresses
2. If the conversation is not found, then create a new conversation with the subject "Policy #12345" and send standard invitation messages to provided email addresses
3. Create a message based on the text provided in *message.txt*
4. Upload the file *attachment.pdf* and set it as an attachment

Updating Conversations

epost Connect™ is also able to post messages into existing conversations. This is done by using the subject and participant emails. The combination of the subject and participant emails have to match exactly that which was provided at the time when the conversation was created.

Based on the example above, the following call will post a message into an existing conversation:

```
-s Policy #12345 -to user1@email.com user2@email.com -m newmessage.txt -att newattachment.pdf
```

The processing is as follows:

1. Search the Owner’s data-set of conversations whose subject is “Policy #12345”, and shared with the user1@email.com and user2@email.com email addresses
2. If the conversation is found then create message based on the text provided in *newmessage.txt*
3. Upload the file *newattachment.pdf* and set it as an attachment

Chapter 6 : Using VTPTrack

The VtpTrack application (VtpTrack) is used to track messages and conversations that have been sent. All sent messages can be tracked, including expired ones. The possible tracking criteria and utility usage are discussed below.

VTPTrack example:

VtpTrack executes in command line style with key-value pairs that are provided in sequence after the executable call. The key value pairs contain information on how the user wishes to filter the results.

The following example illustrates a VTP track command requesting to see the status of all messages recorded that were sent by the *test-app3* account on June 22, 2012.

```
%VTP_HOME%/bin/VtpTrack -a test-app3 -beg 06/22/2012 -end 06/24/2012
```

Note that it is practical to issue these commands from inside the script files (**.cmd** files for DOS type operating systems) which themselves can be executed by double-clicking the file instead of typing a string of commands directly on the command line.

The result produces the following key-value pairs:

Key	Value	Description
a	test-app3	The sender’s epost Connect™ access token for VTP
beg	06/22/2012	The beginning date from which the messages were sent
end	06/24/2012	The ending date from which the messages were sent

Command Line Options

The following table explains the command line parameters, where support options are: M – mandatory, O – Optional:

Key	Command	Support	Default	Description
-a	Account	M	Not Applicable	The account access token associated with this instance of VTP. **Note**: Previous versions of VTP required the username and password of an account. This has now been deprecated to a single access token. Therefore, pre-existing VTP setups must request a single access token for their account in order to use this VTP package. Example 1: -a test-app1

				Example 2: -a 730d75db4a41434bb60a32
-beg	Begin date	O	The date in which the account was created/registered	The date after which the package must have been created. Format must be mm/dd/yyyy. Example: -beg 06/22/2012
-end	End date	O	Current date	The date before which the package must have been created. Format must be mm/dd/yyyy. Example: -end 06/22/2012
-s	Conversation Subject	O	All subjects	A string or sub-string that must match the subject of conversation. **Note**: Messages from all conversations with the same subject will be tracked**.** Example: -s Policy #123456
-r	Participant	O	All recipients	Participant's email address. Participants can be tracked by email address only. Example: -r user1@email.com
-t	Tracking log file	O	Default tracking logs are written to : 'epost.connect/vtp/logs' and have default name: '*tracking-<date-time>.log*'	The message results will be stored in a tracking log file. By default, the application will generate a file named *tracking-<date-time>.log* in the *epost.connect/vtp/logs* directory; where the 'date-time' is the current date and time of the user's PC terminal. The format of log details are as follows: Message ID, Send timestamp, Conversation subject, Sender's account name, Recipient's email, Message status, Current timestamp, Billing code. Example: -t D:/folder1/folder2/test-track.log Where: **D:/folder1/folder2** is the specified destination of the tracking log file, and **test-track. log** is the desired file name.
-e	Exception log file	O	Default exception logs are written to : 'epost.connect/vtp/logs' and have default name: '*exception-<date-*	All exceptions are written to an exception log file. By default, VTP will generate an exception log file named *exception-<date-time>.log* in the *epost.connect/vtp/logs* directory where

			time>.log’	the 'date-time' is the current date and time of the user’s PC terminal. This key allows the user to select an alternative directory and name for the exception file. The format of exception log details are as follows: Timestamp, Subject, Sender's account name, Recipients email address, Error code, Reason of failure. If there is no exception thrown, the exception log will still be present but will be empty (file size will be 0). Example: -e D:/folder1/folder2/vtp-track-error.log Where: • **D:/folder1/folder2** is the specified destination of the exception log file and **vtp-track-error.log** is the desired file name.
-help	Help	O		Provides help on command syntax.

Mandatory Keys

Account identity is the only mandatory key. If keys specifying search criteria are missing epost Connect™ uses default values.

Illegal Keys

VtpTrack has no illegal key groups. Keys specifying search criteria can be specified in any sequence.

Using VTPReceive

The VtpReceive application (VtpReceive) is used to download messages in bulk via the epost Connect™ server. The VtpReceive application provides the ability to automatically download bulk messages from a Connect account either all at once or based on chosen selection criteria using existing **PosteCS** keys **–a**, **-beg**, **-end**, **-all**, **-tdir**, **-l** and **-e**. To download all messages for an account, or to download a subset of messages based on a date parameter, all parameters are passed directly to the executable.

VtpReceive example:

VtpReceive executes in command line style with key-value pairs that follow the executable call. The following example illustrates a command line VtpReceive call including key-value pairs.

```
%VTP_HOME%/bin/VtpReceive  -a test-app1    -beg 05/01/2013    -end 05/08/2013
```

The result produces the following key-value pairs:

Key	Value	Description
a	test-app1	The sender's epost Connect™ access token for VTP
beg	05/01/2013	The begin date of the search i.e. received messages that were created including and after this date.
end	05/08/2013	The end date of the search i.e. received messages that were created including and prior to this date.

Command Line Options

Key	Command	Default Value	Description
-a	Account identity	Not Applicable	The account access token associated with this instance of VTP. **Note**: Previous versions of VTP required the username and password of an account. This has now been deprecated to a single access token. Therefore, pre-existing VTP setups must request a single access token for their account in order to use this VTP package. Example 1: -a test-app1 Example 2: -a 730d75db4a41434bb60a32
-beg	Begin date	The date in which the account was created/registered	The date after which the package must have been created. Format must be mm/dd/yyyy. (Eastern Daylight/Standard Time).

			Example: -beg 06/22/2012
-end	End date	Current date	The date before which the package must have been created. Format must be mm/dd/yyyy. (Eastern Daylight/Standard Time). Example: -end 06/22/2012
-all	All messages		When this key is used, then all items matching selection criteria will be downloaded, regardless of whether they have been downloaded before by this account or not. If this key is not specified, it will only download new items matching selection criteria which have not been downloaded before. Example: -all
tdir	Target Directory	%VTP_HOME%/tdir	The directory where the files will be saved in the local drive. If the directory does not exist, files will be downloaded to the default target directory. A warning will also appear in the exception log file. Example: -tdir D:/folder1/folder2/
-l	Output log file	Logs are written to : 'epost.connect/vtp /logs' and have default name: *'receive-<date-time>.log'*	The status of all transactions (successful or failure) are written to the log file. By default, it will generate a log file named *receive-<date-time>.log* where the 'date-time' is the current date and time of the user's PC terminal. This key allows the user to output the status log file to a directory other than the default *epost.connect/vtp/logs* directory and have a name other than the default above. Format of log details are as follow: Package ID, Send Timestamp, Received Timestamp, Subject, Sender's account email, Message file name or attachment namestatus. Example: -l D:/folder1/folder2/vtp.log Where: • **D:/folder1/folder2** is the specified destination of the log file and **vtp.log** is the desired log file name.
-e	Error log file	Exception logs are written to : 'epost.connect/vtp /logs' and have default name: *'exception-*	All exceptions are written to an exception log file. By default, VTP will generate an exception log file named *exception-<date-time>.log* in the *epost.connect/vtp/logs* directory where the 'date-time' is the current date and time of the user's PC terminal. This key allows the user to select an

		<date-time>.log'	alternative directory and name for the exception file. The format of exception log contents are as follows: Message Id, Timestamp, Subject, Sender's account email, Error code, Reason of failure. If there is no exception thrown, the exception log will still be present but will be empty (file size will be 0). Example: -e D:/folder1/folder2/vtp-error.log Where: • **D:/folder1/folder2** is the specified destination of the exception log file and **vtp-error.log** is the desired exception log file name.
-help	Help		Provides help information about command line.

Backward Compatibility

The VtpReceive application provides the ability to automatically download bulk messages from a Connect account either all at once or based on chosen selection criteria using existing **PosteCS** keys **–a**, **-beg**, **-end**, **-all**, **-tdir**, **-l** and **-e**. The existing scripts have to refer only to the new executable.

Backward Compatibility allows migrated PosteCS customers to continue using the same interfaces and processes for operations on Connect.

VTP provides default behavior for each executable matching current PosteCS experience as close as possible. However, some new epost Connect™ features cannot be accessed using existing commands. New epost Connect™ clients are encouraged to use new command keys which allow full access to all epost Connect™ operations thus making VTP processing faster and more reliable.

Chapter 8 : Status Log Files

VtpSend, VtpReceive and VtpTrack applications produce a status output log upon execution. The primary purpose of this log is to provide information as to whether the execution was successful with the required message being sent, or the target messages being downloaded via VtpReceive or tracked with VtpTrack.

VTPSend

The status log for VtpSend contains the status of all messages sent during the transaction. The default log file is created in the *epost.connect\vtp\logs* directory and has the *name tracking-<date-time>.log* where <date-time> is a combination of the date and time as read from the platform on which the application is running.

The user can specify the name of this log file and path by using the **-l** key (See Chapter 5 :Using VTPSend).

Log record format:

Message ID|Timestamp|Subject|Owner|Participant|Status

Field	Description
Message ID	Message identity.
Timestamp	The date and time the message was sent.
Subject	Conversation subject also known as message title.
Owner	The e-mail address of the owner of the conversation(s) (sender).
Participant	The e-mail addresses of participant(s) receiving the message. If the conversation is shared with multiple recipients, the status of individual email addresses will be shown on separate lines in the log.
Status	Status of the sent process showing: • Message sent • Message not sent For further detail, refer to Chapter 10 Exit Code session.

Example: Message is added to new or existing conversation shared with user1 and user2

```
51|Wed Nov 21 10:11:00 EST 2012|message de Postes  sent
52|Wed Nov 21 10:11:00 EST 2012|message de Postes  sent
```

VTPTrack

The status/tracking log for VTPTrack returns the current sending status for each message found in the system that fits the criteria specified in the corresponding VtpTrack command. The default log file is created in the *epost.connect\vtp\logs* directory and has the name *tracking-<date-time>.log* where <date-time> is a combination of the date and time as read from the platform on which the application is running.

The user can specify the name of this log file and path by using the **-t** key (See Chapter 6 :Using VTPTrack).

Log record format:

```
Message ID|Send Timestamp|Subject|Owner|Participant|Status|Status Timestamp|Billing
Code
```

Log record fields:

Field	Description
Message ID	Message identity.
Send Timestamp	The date and time the message was sent.
Subject	Conversation subject also known as message title.
Owner	The e-mail address of the owner of conversation(s) (sender).
Participant	The e-mail addresses of participant(s) receiving the message. If the conversation is shared with multiple recipients, the status of individual email addresses will be shown on separate lines in the log.
Status	Message status: • Participant is locked • Delivery failed due to non-existent e-mail account on a valid domain • Participant has been notified but no pick-up has been attempted • Message successfully viewed by the recipient
Status Timestamp	Date and time of the current status.
Billing Code	Always empty. epost Connect™ keeps this field for backward compatibility.

Example 1. The status shows the recipient has already received and viewed the package:

64|Thursday Nov 08 12:10:09 EST 2012|Message with successfully viewed by the recipient.|Thursday Nov 08 13:15:46 EST 2012|

Example 2. The status shows the recipient has been notified with the email notification, pending to receive the package:

```
103|Friday Nov 09 10:24:55 EST 2012|Single message|TestAccount|ngagnon home@yahoo.com|Delivery
failed due to non-existent e-mail account on a valid domain| Friday Nov 09 10:27:54
EST 2012||
```

VTPReceive

The status log for VtpReceive contains the status of the execution of the command. The default log file is created in the *epost.connect\vtp\logs* directory and has the name *receive-<date-time>.log'* where <date-time> is a combination of the date and time as read from the platform on which the application is running.

The user can specify the name of this log file and path by using the **-l** key (See Chapter 5 :VtpReceive).

Log record format:

```
Message ID|Sent Timestamp |Received Timestamp|Subject|Owner|File name|Status
```

Field	Description
Message ID	Message identity.
Sent Timestamp	The date and time the message was sent.
Received Timestamp	The date and time the message was received.
Subject	Conversation subject also known as message title.
Owner	The e-mail address of the owner of conversation(s) (sender).
Message or Attachment File name	The name of the message file or attachment file received.
Status	Status of the receive process showing: • Success • Failure • You have no messages to download For further detail, refer to Chapter 7 Exit Code session.

Example:

```
2981|Mon May 20 20:25:47 EDT 2013|Fri May 31 16:43:17 EDT 2013|Test
conversation|epost.connect1@gmail.com|1677-2981_message.txt|Success|

|Fri May 31 13:52:13 EDT 2013||vtp-account@yyy.ca||You have no messages to download
```

Chapter 9 : Exception Files

Exception logs contain details of a failure to execute a specified command. VtpSend, VtpReceive and VtpTrack will produce exception files in the *epost.connect/vtp/logs* directory after each execution. If the file size is 0, this indicates that the command executed successfully without any exception detected during execution. However, if the file size is non-zero, a failure has occurred and the exception file will contain an error code indicating the reason for failure.

VTP Send

The VtpSend application will produce an exception file (non-zero size) after execution if the command was not successful. The default exception file is created in the *epost.connect\vtp\logs* directory and has the name *exception-<date-time>.log* where <date-time> is a combination of the date and time as read from the platform on which the application is running.

The user can specify the name of this log file and path by using the **-e** key (See Chapter 5 :Using VTPSend).

Log record format:

```
Timestamp|Subject|Owner|Participant|Error Code|Reason
```

Log record fields:

Field	Description
Timestamp	The date and time the package was sent.
Subject	Conversation subject.
Owner	The e-mail address of the owner of conversation(s) (sender).
Participant	The e-mail addresses of participant(s) receiving the message. If the conversation is shared with multiple recipients, the status of individual email addresses will be shown on separate lines in the log.
Error code	The error code, see *Chapter 10 :Error and Exit Codes*.
Reason	The description of the error, as outlined in *Chapter 10 :Error and Exit Codes*.

Example:

```
Mon Dec 03 10:19:58 EST 2012|Testing security parameter  Empty participant list.
Conversation requires at least one participant
```

VTP Track

The VtpTrack application will produce an exception file (non-zero size) after execution if the command was not successful. The default exception file is created in the *epost.connect\vtp\logs* directory and has the name *exception-<date-time>.log* where <date-time> is a combination of the date and time as read from the platform on which the application is running.

The user can specify the name of this log file and path by using the **-e** key (See Chapter 5 :Using VTPSend).

Log record format:

```
Timestamp|Subject|Owner|Participant|Error Code|Reason
```

Log record fields:

Field	Description
Timestamp	The date and time the package was sent.
Subject	Conversation subject.
Owner	The e-mail address of the owner of conversation(s) (sender).
Participant	The e-mail addresses of participant(s) receiving the message. If the conversation is shared with multiple recipients, the status of individual email addresses will be shown on separate lines in the log.
Error code	The error code, see *Chapter 10 :Error and Exit Codes*.
Reason	The description of the error, as outlined in *Chapter 10 :Error and Exit Codes*.

Example: Wrong date range is detected:

```
Tue Dec 04 10:04:13 EST 2012|||517|Begin Date is greater then End date.
```

VTPReceive

The VtpReceive application will produce an exception file (non-zero size) after execution if the command was not successful. The default exception file is created in the *epost.connect\vtp\logs* directory and has the name *exception-<date-time>.log* where <date-time> is a combination of the date and time as read from the platform on which the application is running.

The user can specify the name of this log file and path by using the **-e** key (See Chapter 5 :Using VTPSend).

Log record format:

```
Message ID|Timestamp|Subject|Owner| Error Code|Reason
```

Log record fields:

Field	Description
Message Id	Message Id.
Timestamp	The date and time the message was sent.
Subject	Conversation subject.
Owner	The e-mail address of the owner of conversation(s) (sender).
Error code	The error code, see *Chapter 10 :Error and Exit Codes*.
Reason	The description of the error, as outlined in *Chapter 10 :Error and Exit Codes*.

Example:

```
|Fri May 31 16:20:11 EDT 2013|||519,802|Invalid end date, Invalid exception file
specified, reverted to default.
```

Chapter 10 : Error and Exit Codes

Error Codes

Error	Description
16	Invalid Access token
45	Invalid message expiration days.
46	The message attachment size exceeds 20GB.
48	The message attachment size exceeds the maximum allowed for the user account
50	The total message size exceeds its limits.
93	Message file not found
94	Attachment file not found
95	Invitation file not found
97	Error sending message. Please contact Connect Help Desk
157	The conversation topic exceeds 100 characters.
168	Too many invalid login attempts. Participant has been locked out from this message.
205	Invalid notification options.
209	Unauthorized use of message expiration for this message.
211	Unauthorized use of message confirmation success for this message
212	Unauthorized use of message confirmation failure for this message
213	Unauthorized use of message notification include message option for this message.
214	Unauthorized use of message notification include receive page option for this message.

502	Invalid subject text
503	Invalid conversation password
504	Empty participant list. Conversation requires at least one participant.
517	Begin Date is greater then End date.
518	Invalid Begin Date
519	Invalid End Date
520	Invalid Message Id
521	Invalid target directory
801	Invalid log file specified, reverted to default
802	Invalid exception file specified, reverted to default
803	Invalid target directory file specified, reverted to default
1000	General error, please contact Connect Help Desk.
10051	Invalid security option
10053	The language must be either e f en or fr
10054	The conversation topic cannot be left blank.
10055	The personalized message exceeds 500 characters.
10056	The email in invalid for a participant.
10057	The message must include text note, attachment or both

Correcting an Error for VtpSend

If a batch file or batch files do not run and the executable creates an exception file, you will need to create a new batch file for all of the records that failed and then run the new file.

Exit Codes

Upon execution of either the VTPSend, VtpTrack or VtpReceive application, the following status messages are presented on the command line:

Execution Status messages
Processing was completed successfully
Processing not completed successfully, some error was encountered.

The executable returns following the exit codes are as follows:

Exit Code	Description
0	Successfully processed command line(s)
1	Invalid configuration file or invalid VTP_HOME variable in configuration file
2	Argument incomplete of incorrect
3	Invalid URL specified in the configuration file
4	Invalid proxy settings in the configuration file
5	Problem connecting to web service
6	Invalid log/exception file
7	Invalid Target Directory
8	Other errors, check the log file for details

Chapter 11 : Troubleshooting

Problems & Solutions

Problem	Resolution
Unable to contact the server, contact the Connect Administrator	Before contacting the Connect Help Desk: Make sure your are connected to the internet and that the firewall is open.
The system cannot find the path specified Java 'ClassNotFound'	Make sure the path is set to include both the installed Java Runtime Environment and the "installed directory"/ lib after unpacking VTP.
Log files are not created	Check whether it went to the default log file location: $VTPHome/logs.
Account is locked	After 5 unsuccessful login attempts the account is locked for security reasons. To unlock the account you have to contact the Connect Help Desk.

Chapter 12 : Limitations

1) The sender account must be created before using the Volume Transaction Processing application. The Volume Transaction Processing application does not create accounts.

2) Member accounts will not be able to create billing codes.

3) When creating a conversation using the **-i** key parameter, there is a risk of exposing the secure text found in the body of the message. If there is no argument associated with the **-i** key, the secure text will be used as the "personal message" in the email notification sent to the participant.

4) Please note that when using the **-sec** key, there is still the risk of exposing the secure text found in the body of the message by using the **-i** key with no argument as per item 3 above.

5) VTP cannot:
   - update the list of participants after an initial conversation has been sent. The epost Connect™ application will always either put messages into conversations with matching subject and participants or create a new conversation
   - update participant's privileges
   - update the conversation subject
   - create a conversation in a folder or move an existing conversation to a folder

6) VTP will fail if more than one conversation found by subject and participants is located.

7) VTP truncates message content removing trailing spaces and non-printable characters. VTP will fail if message content is empty and an attachment is not provided.

8) The following is a list of special characters for the most popular operating systems being used today:

   - Windows 7 (Cmd.exe) :
     & < > [ ] { } ^ = ; ! ' + , ` ~ * |

   - Unix C-shell :
     ' " * ? [-] { } ~ & ; ( ) $! < > | # @

   - Linux Bash-shell :
     # ; & . : ' " / \ ! * ? $ ( ) { } [ ] < > | - ~ ^

To include special characters (such as the ones mentioned above) in the conversation subject header, you can enclose the subject with beginning and ending double quotes (" ") or precede that character with the corresponding escape character required by the command shell. Escape character will not be inserted into the subject header.

Note: For VtpSend, there is another option to include special characters in a conversation subject header. By using the **-f** key and putting command data in a text file, one can bypass the complication of special characters being interpreted by the shell. Inside the text file, just enter the special characters without the escape character.

Example:

If you enter the command **-a account -s subjectbegin & subjectend -to**, the command will fail as **&** is used to run multiple commands and assumes **subjectend** is another program to run in Windows.

If you enter the command **-a account -s subjectbegin ^& subjectend -to**, the command will execute and the subject taken is : **subjectbegin & subjectend** as **^** has escaped the meaning of **&**

Similarly, if you enter command such as **-a account -s "subjectbegin & subjectend" -to** the command will execute and the subject taken is : subjectbegin & subjectend when it is enclosed in double quotes.

If you enter the command **-a account -s subjectbegin " subjectend -to** the command may fail or send incorrect data as anything after **"** is treated as part of the subject. If **-s** is placed at the end of the command, it can work. But when placed at the middle any parameter after that is taken into subject.

If you enter the command **-a account -s subjectbegin ^" subjectend -to** the command will succeed but the subject is taken as : **subjectbegin subjectend** as **"** will be dropped.

Note:

- Whatever is submitted with special characters without using the **escape character** in the subject or other command parameters may produce undesirable effects.
- A single double quote (") inside the subject line can cause commands to execute incorrectly. A single double quote (") inserted into the subject line via the **escape character** will not cause problems but will be ignored and dropped from the subject line.
- The use of spaces or special characters (as described above) in path variables within the command line can cause issues with proper execution. Ensure that paths are either named without special. characters and spaces or incorporate the proper escape (^) nomenclature within the command line.
- The preceding and trailing spaces entered in the subject key are removed. Also, multiple spaces between words in the subject will be converted to a single space.

- Security protected messages sent by other conversation owners cannot be downloaded using VtpReceive now.

epost Support The name and location of the log are the same as what is specified by the user. If the user does not specify the name and location; the default file name will be *exception-<date-time>.log* under the current working directory.

Appendix

Usage Examples

Notes:

- Key **-a** is omitted for examples as it the same in all cases.
- Email addresses are replaced with **user1**, **user2** for simplicity.

Usage	Command Line	Condition
Create conversation with no participants	-s Policy #12345	Conversation identified by "Policy #12345" and empty list of participants doesn't exist.
Create conversation with 1 participant without messages	-s Policy #12345 -to user1	Conversation identified by "Policy #12345 – user1" doesn't exist.
Create conversation with 1 participant adding message	-s Policy #12345 -to user1 -m message.txt	Conversation identified by "Policy #12345 – user1" doesn't exist.
Add message to existing conversation	-s Policy #12345 -to user1 -m message.txt	Conversation identified by "Policy #12345 – user1" exists.
Create conversation with 1 participant adding message with expiry	-s Policy #12345 -to user1 -m message.txt –exp 365	Conversation identified by "Policy #12345 – user1" doesn't exist.
Add message with expiry to existing conversation	-s Policy #12345 -to user1 -m message.txt –exp 365	Conversation identified by "Policy #12345 – user1" exists.
Create conversation with blank message having 1 attachment	-s Policy #12345 -to user1 -att attachment.txt	Conversation identified by "Policy #12345 – user1" doesn't exist.
Add blank message with 1 attachment to existing conversation	-s Policy #12345 -to user1 -att attachment.txt	Conversation identified by "Policy #12345 – user1" exists.
Create conversation with message having 1 attachment	-s Policy #12345 -to user1 -m message.txt -att attachment.txt	Conversation identified by "Policy #12345 – user1" doesn't exist.
Add message with 1 attachment to existing conversation	-s Policy #12345 -to user1 -m message.txt -att attachment.txt	Conversation identified by "Policy #12345 – user1" exists.
Change subject of conversation	-s Policy #12345 --mod --s Policy #12348 -to user1	Conversation identified by "Policy #12345 – user1" exists.
to download all	-beg 05/25/2013 -all	this will download messages

messages and attachments since certain date -		and attachments sent on May 25,2013 or after, regardless if they have been downloaded previously
to download new messages and attachments between 2 dates into a designated folder -	-beg 05/25/2013 -end 05/30/2013 -tdir c:/myNewMessagesDirectory	this will download new messages and attachments sent between May 25,2013 and May 30, 2013 into the folder c:/myNewMessagesDirectory
to track status of messages sent between 2 dates (VtpTrack) -	-beg 05/25/2013 -end 05/20/2013	this will show the status of the messages sent between May 25,2013 and May 30, 2013

Configuration Examples

Usage	Configuration Settings	Condition
Connect URL specifies protocol. The secure connection flag is ignored.	ServerName=https://epost.ca/Connect-ws	Secure connection on standard port 443.
	ServerName=http://epost.ca/Connect-ws	Not secure connection on standard port 80.

Connect URL doesn't specify protocol. The final URL depends on the secure connection flag.	ServerName=epost.ca/Connect-ws
	SecureConnection=1
		Secure connection. The final URL is https://epost.ca/Connect-ws

	ServerName=epost.ca/Connect-ws
	SecureConnection=0
		Not secure connection. The final URL is http://epost.ca/Connect-ws

Specify non-standard port for Connect's URL.	ServerName=https://epost.ca/Connect-ws Server Port=8443	Secure connection on specified port 8443. The final URL is https://epost.ca:8443/Connect-ws

	ServerName=epost.ca/Connect-ws
	SecureConnection=1
	Server Port=8443	