SIP Feature Server Administration Guide

Python Scripts

5/9/2025

Contents

Python Scripts

> **Important**
>
> The maintenance scripts, described in this article, can only work with the External Cassandra deployment and require External Cassandra to have an open CQL port. Additionally, there are alternative versions of scripts, located in the **python/util** folder that are available for environments with Embedded Cassandra and/or for migration steps between Embedded and External Cassandra up to version 3.11. Python2 (or embedded jython module) is required to run these alternative scripts.

Supported script functions and parameters

Script location: **python/tools**

Before running the maintenance scripts

- Python3 interpreter is installed and it is up and running to run scripts.
- For running scripts, make sure that the path to folder **python/cassandra** is specified and the Python interpreter can find the Cassandra driver module.
- Before running the scripts, note the following:

  1. Set the environment variable **PYTHONPATH**, for example: **PYTHONPATH= <fs_location_dir>/python**
  2. Run the scripts from **<fs_location_dir>/python folder**
     An example command line looks like:

     ```
     python ./tools/<script_name> <parameters_list>
     ```

Including TLS specific arguments

If you are connecting a Cassandra server in a secured mode (TLS enabled), you must use the **--tls** option in the script command line for all scripts (except few scripts, which require a configuration JSON file for starting and configuring TLS there).

Additionally, add the path to the CA certificate file as an argument of the **--cert** option.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of the target Cassandra database.	CassNode01	Yes

Parameters	Description	Sample	Mandatory
Port (-p)	The CQL port of the target Cassandra database.	9042	Yes
TLS Mode (--tls)	Activates the TLS connection mode.	---tls	Yes
Certificate (--cert)	The full path to the location of SSL certificate file.	--cert /home/user/ssl/certs/ca.pem	Yes

Sample: `python ./script.py -H localhost -p 9042 --tls --cert /home/user/ssl/certs/ca.pem`

Modifying the default keyspace name (optional)

All scripts operate with the column families of the default keyspace name **sipfs**. If you want to define an alternative keyspace name, you can do so by specifying the new keyspace name as an argument of the **--keyspace** command line option.

Sample: `python ./script.py -H localhost -p 9042 --keyspace sipfs_new` where, `sipfs_new` is the new keyspace name.

Saving and restoring content of the column families

The maintenance scripts previously used for backing up a single column (**saveColumnFamily.py**, **restoreColumnFamily.py**) were removed in the CQL version. You can now do the same using the following CQL commands that is comparatively more simpler and efficient:

```
 cqlsh -u <username> -p <password> --keyspace <keyspace> -e "COPY <column_family_name> TO
'<path_to_backup_file>';"
 cqlsh -u <username> -p <password> --keyspace <keyspace> -e "COPY <column_family_name> FROM
'<path_to_backup_file>';"
```

Sample: For example, to backup and restore the **device** column family from the **sipfs** keyspace, use the following command:

```
    saving of the column family content to CSV backup file:
        cqlsh -u user -p pass --keyspace sipfs -e "COPY device TO './backup_devices.csv';"
    restoring of the column family content back from CSV file
        cqlsh -u user -p pass --keyspace sipfs -e "COPY device FROM './backup_devices.csv';"
```

applyMessageRetentionLimits.py

Functions:

- Changes the retention limits of existing voicemails.
- This script can be run in two modes:
  - **information mode** - which only reports on mailboxes and their associated voicemail profiles.
  - **execution mode**- which applies the new or changed retention limits.

- Information mode strives to resolve conflicts (where one mailbox is associated with multiple users or user groups with different profiles) before running in the execution mode.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of the target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of the target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Run Mode (--exec)	Activates the execution mode.	--exec	No
Output (-o)	The file name for the output log file with path.	**./cleanupColumnFamilies.log**	Yes

Sample:

```
python ./applyMessageRetentionLimits.py -H localhost -p 9042 -o
./applyMessageRetentionLimits.log
```

cleanupColumnFamilies.py

Functions: Terminates the configuration database synchronization-related column families to enable Configuration Server data reimport later.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of the target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The file name for the output log file with path.	**./cleanupColumnFamilies.log**	Yes

Sample:

```
python ./cleanupColumnFamilies.py -H localhost -p 9042 --keyspace sipfs_new -o
./cleanupColumnFamilies.log
```

copyKeyspaceColumnFamilies.py

Functions: Copies the content of source keyspace column families to the destination keyspace column families.

Parameters: Use parameters in **copyKeyspaceInput.json** and pass it as an input file. This file exists in the same path where this python script is located.

Parameters	Description	Sample	Mandatory
sourceHost	Host of source Cassandra database.	FsNode01	Yes
sourcePort	The CQL port of source Cassandra database.	9042	Yes
destinationHost	The host name of the destination Cassandra database.	CassNode01	Yes
destinationPort	The CQL port number of the destination Cassandra database.	9042	Yes
sourceKeyspace	The source keyspace name.	sipfs	Yes
destinationKeyspace	The destination keyspace name.	sipfs_new	Yes
excludedCFs	A comma-separated list of column family names to be excluded from copying while running the '**copyKeyspaceColumnFamilies.py** script.	message_bytes, device	No
includedCFs	A comma-separated list of column family names to be copied while running the **copyKeyspaceColumnFamilies.py** script.	message_bytes, device	No
sourceHostUserName	The username of the user accessing source Cassandra.	FSadmin	Yes, if authentication is enabled in the source Cassandra cluster.
sourceHostPassword	The password of the user accessing source Cassandra.	FSadmin	Yes, if authentication is enabled in the source Cassandra cluster.
sourceHostTls	Set this option to `true` when SSL is enabled for the source Cassandra connection.	`true`	Yes, if SSL is enabled in the source Cassandra.
sourceHostCert	The path to the source CA certificate file.	**/home/certs/ca.pem**	Yes, if SSL is enabled in the source Cassandra.
destinationHostUserName	The username of the user accessing	FSadmin	Yes, if authentication is enabled in the

Parameters	Description	Sample	Mandatory
	destination Cassandra.		destination Cassandra cluster.
destinationHostPassword	The password of the user accessing destination Cassandra.	FSadmin	Yes, if authentication is enabled in the destination Cassandra cluster.
destinationHostTls	Set this option to `true` when SSL is enabled for the destination Cassandra connection.	`true`	Yes, if SSL is enabled in the destination Cassandra.
destinationHostCert	The path to destination CA certificate file.	**/home/certs/ca.pem**	Yes, if SSL is enabled in the destination Cassandra.

Sample:

```
python ./copyKeyspaceColumnFamilies.py -i ./copyKeyspaceInput.json -o
./copyKeyspaceContent_`date +%y%m%d-%H:%M`.log
```

copyKeyspaceSchema.py

Functions: Creates a keyspace and its column families in the destination Cassandra cluster copied from the source keyspace.

Parameters:

Use parameters in **copyKeyspaceInput.json** and pass it as input file. This file exists in the same path where this python script is located.

Parameters	Description	Sample	Mandatory
sourceHost	The host name of source Cassandra database.	FsNode01	Yes
sourcePort	The CQL port number of source Cassandra database.	9042	Yes
destinationHost	The host name of destination Cassandra database.	CassNode01	Yes
destinationPort	The CQL port number of destination Cassandra database.	9042	Yes
sourceKeyspace	The name of the source keyspace.	sipfs	Yes
destinationKeyspace	The name of the destination keyspace.	sipfs	Yes
excludedCFs	A comma-separated list of column family names to be excluded from	message_bytes, device	No

Parameters	Description	Sample	Mandatory
	copying while running the **copyKeyspaceColumnFamilies.py** script.		
includedCFs	A comma-separated list of column family names to be copied while running the **copyKeyspaceColumnFamilies.py** script.	message_bytes, device	No
sourceHostUserName	The username of the user accessing source Cassandra.	FSadmin	Yes, if authentication is enabled in the source Cassandra cluster.
sourceHostPassword	The password of the user accessing source Cassandra.	FSadmin	Yes, if authentication is enabled in the source Cassandra cluster.
sourceHostTls	Set this option to `true` when SSL is enabled in the source Cassandra connection.	`true`	Yes, if SSL is enabled in the source Cassandra.
sourceHostCert	The path to source CA certificate file.	**/home/certs/ca.pem**	Yes, if SSL is enabled in the source Cassandra.
destinationHostUserName	The username of the user accessing destination Cassandra.	FSadmin	Yes, if authentication is enabled in the destination Cassandra cluster.
destinationHostPassword	The password of the user accessing destination Cassandra.	FSadmin	Yes, if authentication is enabled in the destination Cassandra cluster.
destinationHostTls	Set this option to `true` when SSL is enabled in the destination Cassandra connection.	`true`	Yes, if SSL is enabled in the destination Cassandra.
destinationHostCert	The path to the destination CA certificate file.	**/home/certs/ca.pem**	Yes, if SSL is enabled in the destination Cassandra.
replicationStrategyClassName	The name of the replication strategy class.	`SimpleStrategy`, `NetworkTopologyStrategy`	No, `SimpleStrategy` is used by default.
replicationOptions	The value of the replication map or replication factor.	`{"usw1": 2}`, `1`	Yes, for `NetworkTopologyStrategy` No, for `SimpleStrategy` , (1 is set by default)

Replication strategy class and factor settings:

Class	Replication factor	Value Description
SimpleStrategy	replication_factor : N	Assign the same replication

Class	Replication factor	Value Description
		factor to the entire cluster.
NetworkTopologyStrategy	datacenter_name : N	Assign replication factors to each data center in a comma-separated list.

Sample:

```
python ./copyKeyspaceSchema.py -i ./copyKeyspaceInput.json -o ./copyKeyspaceSchema_`date
+%y%m%d-%H:%M`.log
```

exportMe.py

Functions:

Based on request, this script exports voicemail data in the **.wav** format from the Cassandra database in a client-understandable format.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (--dbhost)	The host name of target Cassandra database.	CassNode01	Yes
Port (--dbport)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (--fileLocation)	Input **.json** file name and its location.	**./in/export_me**	Yes
Output (--outputLocation)	Output result file name and its location.	**./out/export_me**	Yes

Sample:

```
python ./exportMe.py --dbhost localhost --dbport 9042 --fileLocation ./in/export_me --
outputLocation ./out/export_me
```

The following is a sample input JSON file:

```
{
  "caseid": "123456789",
  "consumers": [
    {
      "consumer": [
        { "name": "John Doe" },
        { "name": "John Q. Doe" },
        { "phone": "55551011" }
      ]
    },
    {
      "consumer": [
        { "name": "Dan Akroyd" },
```

```
        { "phone": "55551012" },
        { "phone": "555556162" },
        { "email": "danny@hollywood.com" },
        { "email": "funnyguy@comedy.org" },
        { "fbid": "Dan Akroyd" }
      ]
    }
  ],
  "gim-attached-data": { "kvlist": ["AcctNum", "SSN"] }
}
```

exportVoicemail.py

Functions:

To export Voicemail Mailbox Data, Greetings, Messages, and Message metadata.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output(-o)	Output folder name. It exports the following items: • Mailbox and mailmessage metadata - Export_Voicemail.json • Mail message wav files - messageid.wav file • Greeting wav file (if greeting is set)	**./exportVoicemail**	Yes
agent/agentgroup (--agent)	The export is done only for particular agent or agent group. The list should be provided as a **.csv** file.	**agents_agentgroups.csv**	No
TLS (--tls)	Enable if TLS is used.	NA	No

Sample:

To export all mailboxes, following is the sample script:

```
python exportVoicemail.py -H localhost -p 9042 -o voicemailExport
```

The following is a sample output JSON file:


```
{
    "1115": {
        "mailbox_settings": {
            "activeGreetingType": "Standard",
            "canRetrieve": "true",
            "depositState": "",
            "isAnswerOnlyOn": "false",
            "isEnrollmentDone": "false",
            "lastInvalidLoginTimestamp": "0",
            "lastLoginTimestamp": "0",
            "locale": "",
            "loginAttemptCount": "0",
            "maxMsgCount": "",
            "optoutPhone": "",
            "password": "",
            "reset": "false",
            "salt": "",
            "storageVersion": "5",
            "tenantDbid": "0",
            "timeZoneName": "",
            "unlockedTimestamp": "0"
        },
        "mailbox_greetings": {},
        "voicemail_messages": {
            "7617d1e3-e978-4457-8c9e-ae4d7ffe34d6": {
                "callerid": "799005",
                "callermailboxid": "799005",
                "duration": "2000",
                "fn": "1115-1713097934551.wav",
                "isnew": "true",
                "isprivate": "false",
                "priority": "MediumPriority",
                "retLimit": "",
                "storageVersion": "5",
                "timestamp": "1713097934740",
                "7617d1e3-e978-4457-8c9e-ae4d7ffe34d6": "7617d1e3-e978-4457-8c9e-
ae4d7ffe34d6.wav"
            }
        }
    },
    "111": {
        "mailbox_settings": {
            "activeGreetingType": "Standard",
            "canRetrieve": "true",
            "isAnswerOnlyOn": "false",
            "isEnrollmentDone": "false",
            "lastInvalidLoginTimestamp": "0",
            "lastLoginTimestamp": "0",
            "loginAttemptCount": "0",
            "reset": "false",
            "storageVersion": "5",
            "tenantDbid": "0",
            "unlockedTimestamp": "0"
        },
        "mailbox_greetings": {},
        "voicemail_messages": {
            "18e9f80b-41b2-4d71-93b8-a8291260b0cf": {
                "callerid": "799005",
                "callermailboxid": "799005",
                "duration": "2000",
                "fn": "111-1713097566464.wav",
```

```
                "isnew": "true",
                "isprivate": "false",
                "priority": "MediumPriority",
                "storageVersion": "5",
                "timestamp": "1713097567819",
                "18e9f80b-41b2-4d71-93b8-a8291260b0cf":
"18e9f80b-41b2-4d71-93b8-a8291260b0cf.wav"
            }
        }
    }
}
```

To export specific agent/agent group mailboxes, following is the sample script:

```
python exportVoicemail.py -H localhost -p 9042 -o voicemailExport --agents
agents_agentgroups.csv
```

The following is a sample input csv file:

```
Agent,Agent_Group
2345,
7894,
```

(Agent id 2345 has 111 mailbox, Agent id 7894 has 5687 mailbox)

The following is a sample output JSON file:

```
{
    "111": {
        "mailbox_settings": {
            "activeGreetingType": "Standard",
            "canRetrieve": "true",
            "isAnswerOnlyOn": "false",
            "isEnrollmentDone": "false",
            "lastInvalidLoginTimestamp": "0",
            "lastLoginTimestamp": "0",
            "loginAttemptCount": "0",
            "reset": "false",
            "storageVersion": "5",
            "tenantDbid": "0",
            "unlockedTimestamp": "0"
        },
        "mailbox_greetings": {},
        "voicemail_messages": {
            "18e9f80b-41b2-4d71-93b8-a8291260b0cf": {
                "callerid": "799005",
                "callermailboxid": "799005",
                "duration": "2000",
                "fn": "111-1713097566464.wav",
                "isnew": "true",
                "isprivate": "false",
                "priority": "MediumPriority",
                "storageVersion": "5",
                "timestamp": "1713097567819",
                "18e9f80b-41b2-4d71-93b8-a8291260b0cf":
"18e9f80b-41b2-4d71-93b8-a8291260b0cf.wav"
            }
        }
    },
    "5687": {
        "mailbox_settings": {
            "activeGreetingType": "Standard",
```

```
            "canRetrieve": "true",
            "depositState": "",
            "isAnswerOnlyOn": "false",
            "isEnrollmentDone": "false",
            "lastInvalidLoginTimestamp": "0",
            "lastLoginTimestamp": "0",
            "locale": "",
            "loginAttemptCount": "0",
            "maxMsgCount": "",
            "optoutPhone": "",
            "password": "",
            "reset": "false",
            "salt": "",
            "storageVersion": "5",
            "tenantDbid": "0",
            "timeZoneName": "",
            "unlockedTimestamp": "0"
        },
        "mailbox_greetings": {},
        "voicemail_messages": {
            "93e1fc1c-9ecf-44ec-a8f4-31508822ff3e": {
                "callerid": "799005",
                "callermailboxid": "799005",
                "duration": "2000",
                "fn": "5687-1713098709033.wav",
                "isnew": "true",
                "isprivate": "false",
                "priority": "MediumPriority",
                "retLimit": "",
                "storageVersion": "5",
                "timestamp": "1713098709192",
                "93e1fc1c-9ecf-44ec-a8f4-31508822ff3e": "93e1fc1c-9ecf-44ec-
a8f4-31508822ff3e.wav"
            }
        }
    }
}
```


exportVoicemailSettings.py

Functions:

Exports the Voicemail settings into a CSV file.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	Output folder name	**./exportVoicemailSettings**	Yes

Sample:

```
python ./exportVoicemailSettings.py -H localhost -p 9042 -o ./exportVoicemailSettings
```

forgetMe.py

Functions:

- Deletes the voicemail data of a customer when requested.
- Run this script on the master Feature Server instance to delete the voicemail data. The customer-related information received from the common Web UI will be transformed into a JSON input file.
- The **forgetMe.py** script will fetch the JSON files (that were added since the last execution time to fetch the ANI) added to the **gdpr-directory** option configured in the **[gdpr]** section of the master Feature Server.
- The script will then set the expiry duration to 21 days (by default) for voicemails that correspond to the ANI obtained. The voicemails will be deleted after the expiry duration.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (--dbhost)	The host name of target Cassandra database.	CassNode01	Yes
Port (--dbport)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Expiration Time (--expirationTime)	The voicemail data expiration time (seconds).	1814400	No
Input (--fileLocation)	Input JSON files location folder name.	**./forget_files**	Yes

Sample:

```
python ./forgetMe.py --dbhost localhost --dbport 9042 --fileLocation ./forgetMeFiles
```

The following is a sample input JSON file:

```
{
  "caseid": "123456789",
  "consumers": [
    {
      "consumer": [
        { "name": "John Doe" },
        { "name": "John Q. Doe" },
        { "phone": "55551011" }
      ]
    },
    {
      "consumer": [
        { "name": "Dan Akroyd" },
        { "phone": "55551012" },
        { "phone": "555556162" },
```

```
        { "email": "danny@hollywood.com" },
        { "email": "funnyguy@comedy.org" },
        { "fbid": "Dan Akroyd" }
      ]
    }
  ],
  "gim-attached-data": { "kvlist": ["AcctNum", "SSN"] }
}
```

getAllMailboxCountersInfo.py

Functions: Checks the mailbox counters.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Output (-o)	The name of the output file and its location.	**./getAllMailboxCountersInfo.log**	Yes

Sample:

```
python ./getAllMailboxCountersInfo.py -H localhost -p 9042 -o ./getAllMailboxCountersInfo.log
```

Before running the script verify the following:

- All Feature Servers are up and running.
- There is no voicemail activity (no one can deposit, read, or listen to voicemail).
- Then run the **getAllMailboxCountersInfo.py** script.

getColumnFamilyContent.py

Functions: Retrieves a list of rows (with particular columns content) from the target column family.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Column Family (-c)	The column name of the target column family.	device	Yes

Parameters	Description	Sample	Mandatory
Columns List (-k)	A list of target columns.	USER,NUMBER	No
JSON File (-j)	The file name of the resultant JSON file and its location.	**./results.json**	No
Output (-o)	The name of the output file and its location.	**./getColumnFamilyContent.log**	Yes

Sample:

```
python ./getColumnFamilyContent.py -H localhost -p 9042 -o ./getColumnFamilyContent.log -c
user -k ROLES,RESYNC_ID
```

getColumns.py

Functions: Retrieves the defined columns' values from the target column family.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Column Family (-f)	The column name of target column family.	user	Yes
Columns List (-c)	A list of target columns.	DB_VERSION,ROLES	Yes
Output (-o)	The name of the output file and its location.	**./getColumns.log**	Yes

Sample:

```
python ./getColumns.py -H localhost -p 9042 -o ./getColumns.log -f user -c DB_VERSION,ROLES
```

getMailboxCountAndLastLoginTime.py

Functions: Retrieves the list of mailboxes with assigned and last login date/time info.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra	9042	Yes

Parameters	Description	Sample	Mandatory
	database.		
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
JSON File (-j)	The file name of the resultant JSON file and its location.	**./results.json**	No
Output (-o)	The name of the output file and its location.	**./getMailboxCountAndLastLoginTime.log**	Yes

Sample:

```
python ./getMailboxCountAndLastLoginTime.py -H localhost -p 9042 -o
./getMailboxCountAndLastLoginTime.log
```

getUserKeys.py

Functions: Retrieves the list of users keys.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The name of the output file and its location.	**./getUserKeys.log**	Yes

Sample:

```
python ./getUserKeys.py -H localhost -p 9042 -o ./getUserKeys.log
```

getUsers.py

Functions: Retrieves the list of users containing a column with a particular value.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes

Parameters	Description	Sample	Mandatory
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Column Key (-k)	Target column key.	DB_VERSION	Yes
Column Value (-v)	Target column value.	1	Yes
Output (-o)	The name of the output file and its location.	**./getUserKeys.log**	Yes

Sample:

```
python ./getUsers.py -H localhost -p 9042 -k DB_VERSION -v 1 -o ./getUsers.log
```

importVoicemail.py

Functions: To import Voicemail Mailbox Data, Greetings, Messages, and Message metadata.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
input directory (-i)	Directory where all the **.wav** files for voicemail messages are placed and the exported JSON file.	voicemailExport	Yes
Output directory (-o)	Directory where the execution log is stored.	importVoicemailReport	Yes
TLS(--tls)	Enable if TLS is used.	NA	No

Sample:

```
python importVoicemail.py -H localhost -p 9042 -i voicemailExport -o importVoicemailReport
```

importVoicemailSettings.py

Functions: Imports the Voicemail settings from the CSV file.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes

Parameters	Description	Sample	Mandatory
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	The name of the input folder where the CSV file is located.	**./inVoicemailSettings**	Yes
Output (-o)	The name of the output folder.	**./outVoicemailSettings**	Yes

Sample:

```
python ./importVoicemailSettings.py -H localhost -p 9042 -i ./importVoicemailSettings -o
./outputLogs
```

removeExpiredMessages.py

Functions: Removes expired messages.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Output (-o):	The name of the output file and its location.	**./removeExpiredMessages.log**	Yes

Sample:

```
python ./removeExpiredMessages.py -H localhost -p 9042 -o ./removeExpiredMessages.log
```

Before running the script:

- Ensure that all Feature Servers are up and running.
- Then run **removeExpiredMessages.py** script.

removeSwitchDescription.py

Functions: Removes the Switch description data from the database.

Parameters: Use parameters in **removeSwitchDescription.json** and pass it as an input file. This file exists in the same location where this script is located.

Parameters	Description	Sample	Mandatory
host	The host name of target Cassandra database.	FsNode01	Yes
port	The CQL port number of target Cassandra database.	9042	Yes
keyspace	The name of target keyspace.	sipfs	Yes
switchName	The name of target Switch.	SIP_Cluster	Yes
userName	The username of the user accessing the target Cassandra database.	FSadmin	Yes, if authentication is enabled in the Cassandra cluster.
password	The password of the user accessing the target Cassandra database.	FSadmin	Yes, if authentication is enabled in the Cassandra cluster.
hostTls	Set this option to `true` when SSL is enabled for the target Cassandra connection	`true`	Yes, if SSL is enabled for the Cassandra.
hostCert	The path to the Cassandra host CA certificate file.	**/home/certs/ca.pem**	Yes, if SSL is enabled for Cassandra.

Sample:

```
python  ./removeSwitchDescription.py -i ./removeSwitchDescription.json -o
./removeSwitchDescription.log
```

restoreDeviceCallingProfiles.py

Functions: Restores the Device Calling Profiles data that are not stored in Configuration Server (data saved by corresponding backup script).

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	The name of the input backup CSV file and its location.	**./savedDeviceCallingProfiles.csv**	**Yes**

Parameters	Description	Sample	Mandatory
Output (-o)	The name of the output file and its location.	**./restoreDeviceCallingProfiles.log**	Yes

Sample:

```
python ./restoreDeviceCallingProfiles.py -H localhost -p 9042  -i
./savedDeviceCallingProfiles.csv -o ./restoreDeviceCallingProfiles.log
```

The following is a sample input CSV file:

```
Device ID           Calling Profile ID                              * Note: The header row
should not present in the input CSV file content
------------------------------------------------------------
"20001@SwitchSA02", "24e06da6-1dd3-479a-a0a2-0db2c9aa767c"
"10001@SwitchSA01", "a4ea866b-7dcc-4da6-97ae-24cb14f2e150"
```

restoreUserCallingProfiles.py

Functions: Restores the User Calling Profiles data that are not stored in Configuration Server (data saved by corresponding backup script).

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	The name of the input backup CSV file and its location.	**./savedUserCallingProfiles.csv**	Yes
Output (-o)	The name of the output file and its location.	**./restoreUserCallingProfiles.log**	Yes

Sample:

```
python ./restoreUserCallingProfiles.py -H localhost -p 9042 -i ./savedUserCallingProfiles.csv
-o ./restoreUserCallingProfiles.log
```

The following is a sample input CSV file:

```
User name   User ID Calling                                  Profile
ID                                    * Note: The header row should not present in the
input CSV file content
-------------------------------------------------------------------------------------------------
"un00002",  "57427@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"0758d5a6-355a-4e13-9ef2-63884f88a99c"
"un00001",  "57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"e969a3be-6337-4041-8a9ca270843c6529"
```

restoreUsergroupVmProfiles.py

Functions: Restores the Usergroup Voicemail Profiles data that are not stored in Configuration Server (data saved by corresponding backup script).

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	The name of the input backup CSV file and its location.	**./savedUsergroupVmProfiles.csv**	Yes
Output (-o)	The name of the output file and its location.	**./restoreUsergroupVmProfiles.log**	Yes

Sample:

```
python ./restoreUsergroupVmProfiles.py -H localhost -p 9042 -i ./savedUsergroupVmProfiles.csv
-o ./restoreUsergroupVmProfiles.log
```

The following is a sample input CSV file:

```
Group name  Group ID                                        Voicemail Profile
ID                            * Note: The header row should not present in the input CSV file
content
-------------------------------------------------------------------------------------------------------
"ag002",    "19613@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"92a5fd17-2b4b-493d-8727-004625e0a112"
"ag002",    "19613@f521b229-f599-47d4-81fd-2fbf15b02809",
"11451ec2-d68a-4425-98ebfbf22a24fc7a"
"ag001",    "19612@f521b229-f599-47d4-81fd-2fbf15b02809",
"92a5fd17-2b4b-493d-8727-004625e0a112"
```

restoreUserRoles.py

Functions: Restores the User Roles data that are not stored in Configuration Server (data saved by corresponding backup script).

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes

Parameters	Description	Sample	Mandatory
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	The name of the input backup CSV file and its location.	**./savedUserRoles.csv**	Yes
Output (-o)	The name of the output file and its location.	**./restoreUserRoles.log**	Yes

Sample:

```
python ./restoreUserRoles.py -H localhost -p 9042 -i ./savedUserRoles.csv -o
./restoreUserRoles.log
```

The following is a sample input CSV file:

```
User name   User ID                                          Assigned
roles                                        * Note: The header row should not present in
the input CSV file content
-------------------------------------------------------------------------------------------------------------
"un00001",  "57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"User,Administrator,GroupMailboxAdministrator"
"un00003",  "57428@dcc7a7ac-626a-40c7-b805-e14b71d438d9",   "User,GroupMailboxAdministrator"
"un00002",  "57427@dcc7a7ac-626a-40c7-b805-e14b71d438d9",   "User,GroupMailboxAdministrator"
```

restoreUserVmProfiles.py

Functions: Restores the User Voicemail Profiles data that are not stored in Configuration Server (data saved by corresponding backup script).

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	The name of the input backup CSV file and its location.	**./savedUserVmProfiles.csv**	Yes
Output (-o)	The name of the output file and its location.	**./restoreUserVmProfiles.log**	Yes

Sample:

```
python ./restoreUserVmProfiles.py -H localhost -p 9042 -i ./savedUserVmProfiles.csv -o
./restoreUserVmProfiles.log
```

The following is a sample input CSV file:

```
User name   User ID                                          Voicemail Profile
ID                          * Note: The header row should not present in the input CSV file
content
-------------------------------------------------------------------------------------------------------
"un00002",  "57427@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"11451ec2-d68a-4425-98ebfbf22a24fc7a"
"un00001",  "57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"92a5fd17-2b4b-493d-8727-004625e0a112"
"un00001",  "57426@f521b229-f599-47d4-81fd-2fbf15b02809",
"11451ec2-d68a-4425-98ebfbf22a24fc7a"
"un00003",  "57428@f521b229-f599-47d4-81fd-2fbf15b02809",
"92a5fd17-2b4b-493d-8727-004625e0a112"
```

saveDeviceCallingProfiles.py

Functions: The Device Calling Profile Assignments script creates a CSV file that contains records for all devices with a Calling Profile assigned to them. This CSV file later serves as the input for the Restoring Device Calling Profile Assignments procedure.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The name of the output file and its location.	**./saveDeviceCallingProfiles.csv**	Yes

Sample:

```
python ./saveDeviceCallingProfiles.py -H localhost -p 9042 -o ./saveDeviceCallingProfiles.csv
```

The following is a sample output CSV file:

```
Device ID               Calling Profile ID
"20001@SwitchSA02",     "24e06da6-1dd3-479a-a0a2-0db2c9aa767c"
"10001@SwitchSA01",     "a4ea866b-7dcc-4da6-97ae-24cb14f2e150"
Every record of the device calling profile in the CSV file contains a device ID and the
corresponding ID of a Calling Profile assigned to the device.
Device ID consists of corresponding DN number and the switch name the device belongs to,
separated by ‘@’; for instance: 10001@SwitchSA01.
```

saveUserCallingProfiles.py

Functions:

The User Calling Profile Assignments script creates a CSV file that contains records of all Users with a Calling Profile assigned to them. The CSV file later serves as an input for the Restoring User Calling Profile Assignments procedure.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The name of the output backup CSV file and its location.	**./saveUserCallingProfiles.csv**	Yes

Sample:

```
python ./saveUserCallingProfiles.py -H localhost -p 9042 -o ./saveUserCallingProfiles.csv
```

The following is a sample output CSV file:

```
User name   User ID                                          Calling Profile ID
"un00002",  "57427@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"0758d5a6-355a-4e13-9ef2-63884f88a99c"
"un00001",  "57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"e969a3be-6337-4041-8a9ca270843c6529"
```

Every record of the user calling profile in the CSV file contains a user name, user ID, and the corresponding ID of a Calling Profile assigned to the user. User ID consists of the corresponding person DBID and Configuration Server GUID separated by '@'; for instance: 57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9

saveUsergroupVmProfiles.py

Functions: The User Group Voicemail Profile Assignments script creates a CSV file that contains records of all User Groups with a Voicemail Profile other than the one assigned to them by System Profile. The CSV file later serves as an input for the Restoring User Group Voicemail Profile Assignments procedure.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The name of the output backup CSV file and its location.	**./saveUsergroupVmProfiles.csv**	Yes

Sample:

```
python ./saveUsergroupVmProfiles.py -H localhost -p 9042 -o ./saveUsergroupVmProfiles.csv
```

The following is a sample output CSV file:

```
Group name  Group ID Voicemail                              Profile ID
"ag002",    "19613@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"92a5fd17-2b4b-493d-8727-004625e0a112"
"ag002",    "19613@f521b229-f599-47d4-81fd-2fbf15b02809",
"11451ec2-d68a-4425-98ebfbf22a24fc7a"
"ag001",    "19612@f521b229-f599-47d4-81fd-2fbf15b02809",
"92a5fd17-2b4b-493d-8727-004625e0a112"
```

Every record of the user group voicemail profile in the CSV file contains a group name, group ID and corresponding ID of a Voicemail Profile assigned to the user group. User group ID consists of the corresponding Agent Group DBID and Configuration Server GUID separated by '@'; for instance: 19613@dcc7a7ac-626a-40c7-b805-e14b71d438d9

saveUserRoles.py

Functions: The User Feature Server Roles script creates a CSV file that contains records of all users with Roles different from the default User role. The CSV file later serves as an input for the Restoring User Roles procedure.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The name of the output backup CSV file and its location.	**./saveUserRoles.csv**	Yes

Sample:

```
python ./saveUserRoles.py -H localhost -p 9042 -o ./saveUserRoles.csv</tt>
```

The following is a sample output CSV file:

```
User name   User ID                                        Assigned roles
"un00001",  "57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"User,Administrator,GroupMailboxAdministrator"
"un00003",  "57428@dcc7a7ac-626a-40c7-b805-e14b71d438d9",   "User,GroupMailboxAdministrator"
"un00002",  "57427@dcc7a7ac-626a-40c7-b805-e14b71d438d9",   "User,GroupMailboxAdministrator"
```

Every record of the user roles in the CSV file contains user name, user ID, and the corresponding set of roles assigned to the user. User ID consists of the corresponding person DBID and Configuration Server GUID separated by '@'; for instance: 57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9

saveUserVmProfiles.py

Functions: The User Voicemail Profile Assignments script creates a CSV file containing records of all users with a Voicemail Profile other than the one assigned to them by System Profile. The CSV file later serves as an input for the Restoring User Voicemail Profile Assignments procedure.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Output (-o)	The name of the output backup CSV file and its location.	**./saveUserVmProfiles.csv**	Yes

Sample:

```
python ./saveUserVmProfiles.py -H localhost -p 9042 -o ./saveUserVmProfiles.csv
```

The following is a sample output CSV file:

```
User name User ID Voicemail Profile ID
"un00002", "57427@dcc7a7ac-626a-40c7-b805-e14b71d438d9", "11451ec2-d68a-4425-98ebfbf22a24fc7a"
"un00001", "57426@dcc7a7ac-626a-40c7-b805-e14b71d438d9",
"92a5fd17-2b4b-493d-8727-004625e0a112"
"un00001", "57426@f521b229-f599-47d4-81fd-2fbf15b02809", "11451ec2-d68a-4425-98ebfbf22a24fc7a"
"un00003", "57428@f521b229-f599-47d4-81fd-2fbf15b02809",
"92a5fd17-2b4b-493d-8727-004625e0a112"
```

Every record of the User Voicemail Profile in the CSV file contains a user name, user ID, and the corresponding ID of a Voicemail Profile assigned to the user. User ID consists of the corresponding person DBID and Configuration Server GUID separated by '@'; for instance: 57426@cb2fdedda57f-49e6-a54d-3f930eb1dfc5

set_mailbox_user_tz.py

Functions: Updates the time zones of mailboxes and the time zones of users who are associated with the mailbox.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes

Parameters	Description	Sample	Mandatory
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Run Mode (--exec)	Activates the execution mode.	--exec	No
Input (-i)	Input TimeZones CSV file name.	**./mailbox_user_tz.csv**	Yes
Output (-o)	The name of the output file and its location.	**./set_mailbox_user_tz.log**	Yes

Sample:

```
python ./set_mailbox_user_tz.py -H localhost -p 9042 --exec -i ./mailbox_user_tz.csv -o
./set_mailbox_user_tz.log</tt>
```

The following is a sample input CSV file:

```
Mailbox Number      Time Zone ID            * Note: The header row should not present in the
input CSV file content
---------------------------------------
7100,               America/New_York
7200,               America/Los_Angeles
```

setAdminRoles.py

Functions: Sets administrative roles for users from the list.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Input (-i)	Input Users CSV file name.	**./setAdminRoles.csv**	Yes
Output (-o)	The name of the output file and its location.	**./setAdminRoles.log**	Yes

Sample:

```
python ./setAdminRoles.py -H localhost -p 9042 -i ./setAdminRoles.csv -o ./setAdminRoles.log
```

The following is a sample input CSV file:

```
User Key        * Note: The header row should not present in the input CSV file content
--------------
TServ.User1
TServ.User2
```

TServ.User3

updateMailboxMessagesCFFormat.py

Functions: Updates the Mailbox Message CF record format.

Parameters:

Parameters	Description	Sample	Mandatory
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No
Format (--cassandracounter)	Updating of the CF record to old format: --cassandracounter true new format: --cassandracounter false	false	Yes
Output (-o)	The name of the output file and its location.	**./updateCFFormat.log**	Yes

Sample:

```
python./updateMailboxMessagesCFFormat.py -H localhost -p 9042  --cassandracounter false -o
./updateMailboxMessagesCFFormat.log
```

The following is a sample format:

```
New Record Format:
<tt>KoMailbox,  a38c7205-fa51-495c-9dd0-d70b2c2c4155,   {"is_new": "y", "is_high_priority":
"n"}</tt>

Old Record Format:
<tt>KoMailbox,  a38c7205-fa51-495c-9dd0-d70b2c2c4155,   mailbox-5108-message-1</tt>
```

voicemail-core

corruptMessagesInMailboxmessages.py

This scripts replaces the following scripts:

- checkMailboxMessagesIntegrity.py
- cleanupMailboxMessagesCF.py
- getAllProblematicMailboxes.py

Functions: To discover and fix corrupted messages in mailboxMessages CF. MessageId present in

mailboxMessages CF but not present in mailmessage and message_bytes CF, this occurs when deleted messages reappear.

Parameters:

Parameters	Description	Sample	Mandatory	Default Value
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes	NA
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes	NA
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No	sipfs
Output (-o)	The output JSON file with corrupted messages list.	corruptMessagesInMailboxmessages	Yes	NA
Fix (-f)	If `-f` is not given, the script only finds the corrupted messages If `-f` is enabled, the script finds the corrupted messages and fix them.	NA	No	Fix disabled.
Consistency (-c)	Read and Write Consistency Levels.	LOCAL_QUORUM	No	ONE

Sample:

```
python corruptMessagesInMailboxmessages.py -H localhost -p 9042 -o
corruptMessagesInMailboxmessages
```

The following is a sample output JSON file:

```
{
  "1001": [
    "c007c70b-7431-45ec-974a-617722bf7b01",
    "928e95aa-9d4a-4b8c-96af-5097c61460ae"
  ]
}
```

corruptMessagesInMailboxmessagesAndMailMessage.py

This script replaces the following scripts:

- problematicMailMessages.py
- cleanupMailmessageCF.py

Functions:

To discover and fix corrupted messages in mailboxMessages CF and mailmessage CF. MessageId present in mailboxMessages and mailmessage CF but not present in message_bytes CF and retention limit not set for message bytes, this occurs when deleted messages reappear.

Parameters:

Parameters	Description	Sample	Mandatory	Default Value
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes	NA
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes	NA
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No	sipfs
Output (-o)	The output JSON file with corrupted messages list.	corruptMessagesInMailboxmessages	Yes	NA
Fix (-f)	If -f is not given, the script only finds the corrupted messages If -f is enabled, the script finds the corrupted messages and fix them.	NA	No	Fix disabled
Consistency (--consistency)	Read and Write Consistency levels.	LOCAL_QUORUM	No	ONE

Sample:

```
python corruptMessagesInMailboxmessagesAndMailMessage.py -H localhost -p 9042 -o
corruptMessagesInMailboxmessagesAndMailMessage
```

The following is a sample output JSON file:

```
{
  "1001": [
    "c007c70b-7431-45ec-974a-617722bf7b01",
    "928e95aa-9d4a-4b8c-96af-5097c61460ae"
  ]
}
```

countMailboxes.py

Functions: Gets the total number of mailboxes that a tenant uses, based on mailboxmessages CF.

Parameters:

Parameters	Description	Sample	Mandatory	Default Value
Host Name (-H)	The host name of	CassNode01	Yes	NA

Parameters	Description	Sample	Mandatory	Default Value
	target Cassandra database.			
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes	NA
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No	sipfs
Output (-o)	The output log contains the total mailbox count.	countMailboxes	Yes	NA

Sample:

```
python countMailboxes.py -H localhost -p 9042 -o countMailboxes
```

removeMailboxAndMailboxMessages.py

Functions: Deletes all mailboxes and its messages based on the given input. Remove the mailbox option (TServer→gvm_mailbox) from Agent or Agent Group before running this script.

Parameters:

Parameters	Description	Sample	Mandatory	Default Value
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes	NA
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes	NA
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No	sipfs
Input (-i)	Input Users CSV file with mailbox id to be removed.	mailbox_list.csv	Yes	NA
Output (-o)	The output log file contains details of deleted mailboxes and its messages.	removeMailboxAndMailboxMessages	Yes	NA
Consistency (--consistency)	Read and Write Consistency levels.	LOCAL_QUORUM	No	ONE

Sample:

```
python removeMailboxAndMailboxMessages.py -H localhost -p 9042 -o
removeMailboxAndMailboxMessages -i mailbox_list.csv
```

The following is a sample input CSV file:

```
799033
1001
```

removeMailboxMessages.py

Functions: Deletes all messages from the provided input mailbox list.

Parameters:

Parameters	Description	Sample	Mandatory	Default Value
Host Name (-H)	The host name of target Cassandra database.	CassNode01	Yes	NA
Port (-p)	The CQL port number of target Cassandra database.	9042	Yes	NA
Keyspace (--keyspace)	The name of target keyspace.	sipfs	No	sipfs
Input (-i)	Input Users CSV file with their mailbox ID.	mailbox_list.csv	Yes	NA
Output (-o)	The output log file contains details of deleted messages.	removeMailboxMessages	Yes	NA
Consistency (--consistency)	Read and Write Consistency levels.	LOCAL_QUORUM	No	ONE

Sample:

```
python removeMailboxMessages.py -H localhost -p 9042 -o removeMailboxMessages -i
mailbox_list.csv
```

The following is a sample input CSV file:

```
799033
1001
```