NewNet Mobile Messaging Tools R04.10.04

Operator Manual

Release 17.4 Revision A
February 2019

Table of Contents

Chapter 1

Introduction

Topics:

1.1 About this Document

This document contains all relevant details required for the operation of the command-line tools included in the NewNet Mobile Messaging Tools package and in other NewNet Mobile Messaging components, such as the Router and HUB.

Tools is a collection of command-line tools from the NewNet suite of SS7 messaging products.

This document contains a description of the general operations aspects of Tools. Because the available functions are licensed and depend on the specific implementation, not all functions and/or applications contained in this document may be relevant or applicable to the system you will be working with.

1.2 Scope

This document discusses the functionality of the NewNet Mobile Messaging Tools component.

1.3 Intended Audience

This document is meant for everybody interested in how Tools can best be used, but mainly for:

- **Implementation Engineers** who are responsible for the pre-installation, on-site installation and configuration of NewNet Mobile Messaging components in the end-user environment.
- **Maintenance and Support Engineers** who are responsible for maintaining the total system environment of which NewNet Mobile Messaging components are a part.
- **Network Operators** who are in charge of the daily operation of the NewNet Mobile Messaging systems and infrastructure.

1.4 Documentation Conventions

Typeface or Symbol	Meaning	Example
Bold	Refers to part of a graphical user interface.	Click **Cancel**.
`Courier`	Refers to a directory name, file name, command, or output.	The `billing` directory contains...
`<pointed brackets>`	Serves as a placeholder for text that the user will replace, as appropriate in context.	The file is called `MGRdata.xml.<ip>.gz`, where `<ip>` is the server's IP address.
`[square brackets]`	Indicates an optional command.	`[--validateonly]`

Typeface or Symbol	Meaning	Example
Note:	Indicates information alongside normal text, requiring extra attention.	**Note:** Ensure that the configuration...

\ (Unix)	Denotes line continuation; the character should be ignored as the user types the example, and ENTER should only be pressed after the last line.	`% grep searchkey \`
		`data/*.dat`

1.5 Locate Product Documentation on the Customer Support Site

Access to NewNet's Customer Support site is restricted to current NewNet customers only. This section describes how to log into the NewNet Customer Support site and locate a document. Viewing the document requires Adobe Acrobat Reader, which can be downloaded at www.adobe.com.

1. Log into the NewNet Customer Support site.

   Note: If you have not registered for this new site, click the **Register Here** link. Have your customer number available. The response time for registration requests is 24 to 48 hours.

2. Click the **Product Support** tab.
3. Use the Search field to locate a document by its part number, release number, document name, or document type. The Search field accepts both full and partial entries.
4. Click a subject folder to browse through a list of related files.
5. To download a file to your location, right-click the file name and select **Save Target As**.

Chapter 2

Scripting Command-Line Tools

Topics:

2.1 Introduction

Many command-line interface (CLI) tools are scriptable. You can provide options on the command line or via a standard input (without requiring a terminal), and each tool reports exit codes that can allow a script to react when errors occur.

All tools return exit code:

- 0 upon success
- 1 if an error occurs

For example, `tp_status` returns 0 when all configured processes are running. It returns 1 when one process is not available.

2.2 Limitations

The following command-line tools do not return exit codes and are therefore not scriptable:

- `tp_start`
- `tp_stop`

2.3 Examples

The following example illustrates how display the exit code when executing a tool on the command line.

Command:

```
tp_install_mgr --check; echo "ExitStatus: $?"
```

Output:

```
Current MGR version
R04.03.10.01

Current database settings
host : localhost
port : 3306
user : root
pass : lokal$

Current role setting
Master

Current device versions
AMS : R01.02.00.00
CCI : R01.01.00.00
EMG : Not Installed
FAF : R02.01.00.01
HUB : R04.01.12.00
LGP : R01.03.00.00
PBC : R01.04.31.01
```

```
RTR : R04.01.15.00
STV : R04.03.00
SYS : Not Installed
ExitStatus: 0
```

The following example illustrates how to use the exit code in a script:

```
#!/bin/bash

tp_install_mgr --faulty_option 2>/dev/null

if [ $? ]
then
echo "Something went wrong: $?"
else
echo "All is well"
fi
```

Chapter 3

Command-Line Tools

Topics:

3.1 Introduction

This chapter provides information about command-line interface (CLI) tools.

3.2 tp_backup

The `tp_backup` tool is part of the backup and restore solution for NewNet Mobile Messaging network elements. `tp_backup` will create a backup that can be restored with `tp_restore`.

See also *tp_restore*.

3.2.1 Synopsis

```
tp_backup
```

3.2.2 Options

Option	Description
`-h, -?, --help`	Print the documentation
`-v, --verbose`	Print status information to `STDOUT`
`-t, --temp`	The location to use as a temporary storage (default: `/var/TextPass/backup`)
`-f, --file`	The filename to use for the backup (default: `<hostname><timestamp>`)

3.2.3 Usage

The `tp_backup` script will create a backup using `tar` and `gzip`. It uses the `/usr/TextPass/etc/*.tp_backup` files to decide which files to backup.

The backup follows these steps:

1. Create the temporary stage area
2. Read and parse all the configuration files
3. Execute the pre-backup commands
4. Copy all the files
5. Tar and compress the files
6. Execute all the post-backup commands
7. Cleanup.

During this process detailed information is written to the trace file. This file resides in the same location as the final backup file. The backup file is named `<hostname><timestamp>.tar.gz` by default. Since the files are system files, only the *root* account can perform a backups.

Please note the limitations and refer to the Backup and Restore manual.

3.2.4 Configuration

The backup process is controlled by configuration files. These files are placed in `/usr/TextPass/etc/` and have the extension `.tp_backup`.

The backup configuration files consist of one or more lines. Every line can contain a single statement. A statement must start with:

```
FILE: DIR: PRE_BACKUP: POST_BACKUP: PRE_RESTORE: POST_RESTORE:
```

Where:

`FILE:`	Indicates a single file that needs to be backed up.
`DIR:`	Indicates a directory that needs to be backed up. This directory will be backed up recursively. Links are not followed.
`PRE_BACKUP:`	Commands that will be executed before that backup starts. This can be used to trigger a database dump, or to send a 'HUP' to a process to decouple a log file.
`POST_BACKUP:`	Commands that will be executed after the backup has finished.
`PRE_RESTORE:`	Commands that will be executed before the restore starts.
`POST_RESTORE:`	Commands that will be executed after the restore has finished.

3.2.5 Limitations

Important: The `tp_backup` and `tp_restore` tools are currently for Linux based systems only, for example RHEL7.

It is the operator's responsibility to keep the backup files in a save location.

In case databases are backed up, manual steps may be needed. For example if the node is used in a cluster, the restore procedure may differ.

In case a cluster is backed up; it is vital that the cluster manager node is backed up first. That triggers the backup of the database nodes, which in turn can be backed up.

3.3 tp_config

The `tp_config` tool validates and/or activates the common and/or host-specific semi-static (XML) configuration files.

3.3.1 Synopsis

```
tp_config [--validateonly] [--validatecommonconfig] [<component>] \
[specific-config-file [common-config-file]]
```

3.3.2 Options

Option	Description
`--validateonly`	Validates the configuration files.
`--validatecommonconfig`	Validates the common configuration file.
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.3.3 Operands

Operand	Description
`specific-config-file`	Name of the host-specific configuration file.
`common-config-file`	Name of the common configuration file.

3.4 tp_fclient

The `tp_fclient` tool manages the replication of XML configuration data files from a server. `tp_fclient` enables a client system to subscribe to all changed XML configuration data files that the MGR on the assigned server produces.

3.4.1 Synopsis

```
tp_fclient --continuous --directory=<directory> --timeout=timeout \
--interval=interval --script=script <server1> [<server2>] [<server3>]
```

3.4.2 Options

Option	Description	Default
`--continuous`	Forces the client to continuously attempt to connect to the configured server(s) until a connection is established.	only_once
`--directory`	Overrules the default target directory on the client system. `directory` specifies the directory to which the XML configuration data files will be copied on the client side.	/usr/local/apache/ mBalance/TPManager/ data

Option	Description	Default
`--timeout`	Overrules the default time-out for attempts to connect to the configured server(s). `timeout` specifies the time (in seconds) that the client will wait for a response from the server after making a connection attempt: • `continuous` mode—Initiates a next connect-attempt. • `only_once` mode—Initiates a connect attempt to the next server (if configured) or fails in error. The minimum time-out value is 1 second; the maximum is 60 seconds. **Note:** A long time-out may delay the activation of configuration changes.	3 seconds
`--interval`	Overrules the default polling interval used to check whether an XML configuration data file has changed. `interval` specifies the time (in seconds) that the client will check if an XML file on the server has changed (if the client is connected to the server). The minimum interval value is 1 second; the maximum is 60 seconds.	1 second
`--script`	Specifies a script to execute after one or more XML files have been changed. `script` specifies the name of the script to execute.	Not applicable

3.4.3 Operands

Operand	Description
`server1`	Host name or IP address of the primary server.
`server2`	Host name or IP address of the secondary server.
`server3`	Host name or IP address of the tertiary server.

3.5 tp_filter

Trace filtering is a HUB feature that can capture incoming and outgoing UCP, SMPP, and CIMD messages and MXP traffic. Trace filters are created in the MGR interface. The trace receiver tool can collect the captured trace data and write it to a file (see *tp_trace_receiver*).

The `tp_filter` tool allows you to query the configuration of the trace filter.

`tp_filter` can work with multiple devices simultaneously. For example, executing `tp_filter` with `--consistency` on one HUB will cause `tp_filter` to check the configuration on all HUBs in the same network discovery group.

3.5.1 Synopsis

```
tp_filter --show [--consistency][--device=<host>[:<port>]]
```

```
tp_filter --version
```

```
tp_filter --help
```

3.5.2 Options

Option	Description
`--help`	Provides information about the syntax of `tp_filter`.
`--version` `-v`	Provides the release and version of the trace filter command line interface.
`--show` `-s`	Shows the properties of all configured trace filters and conditions.

3.5.2.1 show

```
tp_filter --show [--consistency][--device=<host>[:<port>]]
```

```
tp_filter -s [--device=<host>[:<port>]]
```

3.5.2.1.1 Description

The `show` option shows the details of all the trace filters and conditions currently configured.

3.5.2.1.2 Parameters

Parameter	Description
`--consistency`	When provided, this parameter enables the consistency check. If consistency is enabled, `show` performs a consistency check on the configurations of all nodes. It will take as base configuration, the configuration on the node with the highest uptime. The results are displayed

Parameter	Description
	for each filter or condition in a column `Conflicts`. In case of conflicts, a list of conflicting nodes is displayed in the `Conflicts` column. If no conflicts exist, `ok` is displayed.
`--device`	Identification of the device.

Note: `show` can be used without any parameters.

3.5.2.1.3 Examples

The following example shows the output when two filters are configured, each with two conditions.

Command:

```
tp_filter -s
```

Output:

```
Filter configuration
====================
Idx     Name            receiver         state
1       mb01            10.0.4.25:51909  active
Conditions
=========
Idx     type    value
1.1     ip      192.168.1.1
1.2     aid     42

Idx     Name            receiver         state
2       mb02            10.0.4.48:50025  inactive
Conditions
=========
Idx     type    value
2.1     app     application1
2.2     sn      1234
```

The following example shows the output in which two filters are configured, each one with two conditions associated to it, when consistency check is enabled.

Command:

```
tp_filter -s --consistency
```

Output:

```
Filter configuration
====================
Idx     Name            receiver         state      Conflicts
1       mb01            10.0.4.25:51909  active     10.0.0.24
Conditions
=========
Idx     type    value           Conflicts
1.1     ip      192.168.1.1     10.0.0.24
1.2     aid     42              10.0.0.24

Idx     Name            receiver         state      Conflicts
2       mb02            10.0.4.48:50025  inactive   ok
Conditions
=========
Idx     type    value           Conflicts
2.1     app     application1    ok
2.2     sn      1234            ok
```

This example depicts a situation where the filter configuration between the 'oldest' node and the device at 10.0.0.24 having different configurations on filter 1 and the associated conditions.

This could have been caused by the scenario where the first filter and conditions are set with both nodes up. Then the node on 10.0.0.24 is rebooted. Next the second filter and conditions are set. In this case, filters 1 and 2 and its conditions exist in the not-rebooted node only, whereas on node 10.0.0.24 only filter 2 exists.

3.6 tp_get

The `tp_get` tool provides the value of a single MIB object.

3.6.1 Synopsis

```
tp_get [<component>] [--device=<host>[:<port>]] <OID>
```

Note:

- The valid scenarios on the device option with IPv6 address will be:

  ```
  --device=<IPv6>
  --device=[IPv6]:<port>
  ```

  Example:

  ```
  tp_get --device=fe80::20c:29ce:ce00:79ef sysDescr.0
  tp_get --device=[fe80::20c:29ce:ce00:79ef]:11161 sysDescr.0
  ```

- The IPv6 address should be enclosed with '[]' when provided with port.

3.6.2 Options

Option	Description
`--device`	Identifies the device by IP address and port number.
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.6.3 Operands

Operand	Description
`OID`	Object-identifier that uniquely identifies the SNMP attribute.

3.7 tp_getnext

The `tp_getnext` tool provides the identifier and value of the next MIB object after the one identified.

3.7.1 Synopsis

```
tp_getnext [<component>] [--device=<host>[:<port>]] <OID>
```

3.7.2 Options

Option	Description
`--device`	Identifies the device by IP address and port number.
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.7.3 Operands

Operand	Description
`OID`	Object-identifier that uniquely identifies the starting SNMP attribute.

3.8 tp_fserver

The `tp_fserver` tool manages the replication of XML configuration data files from a server. `tp_fserver` enables a server system to interact with clients.

3.8.1 Synopsis

```
tp_fserver
```

3.8.2 Options

`tp_fserver` has no options.

3.8.3 Operands

`tp_fserver` has no operands.

3.9 tp_gen_enc_key

The `tp_gen_enc_key` script generates the key file for the SMS Encryption feature. If a file with the same name is already located at given path, it will be backed up with timestamp information. The backup file is named `<file_name>.<timestamp>`.

The script does not re-create the given directory if it already exists, so the directory permissions will not be applicable for this case.

3.9.1 Synopsis

```
tp_gen_enc_key --output_path=<directory> --directory_permission=<0770>
--file_name=<file name> --file_permission=<0640>
```

3.9.2 Options

Option	Description
`--output_path`	Output path of the key file (default: `/usr/TextPass/.crypt`)
`--directory_permission`	Directory permissions of the output path (default: `0770`)
`--file_name`	Name of the key file (default: `ud_crypto_key`)
`--file_permission`	Key file permissions (default: `0640`)

3.10 tp_manage_user

The `tp_manage_user` tool allows management of multiple NMM users on the same server.

With multi-instance feature, multiple NMM users can be added on a NMM server. Each NMM user can run an instance of RTR, HUB, FAF, PBC, SSI, IIW, AMS, Map-Screener, EC-ABM and LGP (if Logging Element).

`tp_manage_user` allows the operator to:

- Create new NMM User and `user.tp_backup` file (See also *tp_backup*).
  - User name will be `tpuserxx`, where xx is user number.
  - Up to 9 NMM Users can be added.
  - New User will be part of the `textpass` group.
  - For each new user, a `user.tp_backup` will be created in `/usr/TextPass/etc`. This file will control the user configuration to be backed up.
- Delete existing users and remove the corresponding `user.tp_backup` files from `/usr/TextPass/etc`.
  - Only NMM users can be deleted using this tool.
  - Default `textpass` user cannot be deleted.
- Display SNMP port information for existing NMM Users.

Note: By default `textpass` user exists and operator can add up to 9 more NMM users.

3.10.1 Synopsis

```
/usr/TextPass/bin/tp_manage_user [--add_user --port=<base_port>
--snmp_identifier=<snmp_id>
/usr/TextPass/bin/tp_manage_user [--delete_user --user_name=<user_name>]
```

```
/usr/TextPass/bin/tp_manage_user [--info]
/usr/TextPass/bin/tp_manage_user [--help]
```

3.10.2 Options

Option	Description
`-a, --add_user`	Creates a new NMM user on a server. Up to 9 NMM users can be added
`-d, --delete_user`	Deletes an existing NMM user **Note:** `textpass` user cannot be deleted.
`-i, --info`	Displays information about all existing NMM users (including `textpass` user)
`-h, -?, --help`	Print the documentation

3.10.3 Operands

Operand	Description
`-p, --port`	The unique base port which will be used to configure SNMP port range for new user

`-snmpid,`
`--snmp_identifier`
	Unique SNMP Identifier used to differentiate SNMP TRAPs raised for new NMM user

`-u, --`user_name	Username to be deleted

1. All options `--add_user`, `--delete_user`, `--info` and `--help` are mutually exclusive.
2. `--port` and `--snmp_identifier` can be specified only with `--add_user` option.
3. Command option `--delete_user` accepts only `--user_name` command operand.
4. Command options `--info` and `--help` do not require any other command operand.

3.10.4 Usage

3.10.4.1 Adding a New NMM User

A new NMM user can be added to the multi-instance setup by executing the following command as `root` user:

```
/usr/TextPass/bin/tp_manage_user [--add_user --port=<base_port>
--snmp_identifier=<snmp_id>
```

Here, `--add_user` command option is provided to specify that a new user is to be added.

`--port` takes the value for the base port (unique for each user) to be entered by the user. This base port will be used to generate all SNMP ports, SNMP trap ports and MXP ports for the new user. The value for all these ports can be seen in the '.textpass' file of the user present at the path "/usr/tpuserxx", where tpuserxx is the newly created user. Following are the limitations for the base port:

- It should contain only digits
- It should range between 1100 - 50000
- 9000 - 15000 is an excluded range as ports lying between this range are already used by textpass user
- Ports less than 1025 are privileged ports and hence, cannot be used
- It should be a multiple of 100
- It should be unique for each NMM user

`tp_manager_user --info` can be used to display SNMP ports information for existing users.

`--snmp_identifier` indicates unique SNMP Identifier used to differentiate SNMP TRAPs raised for new NMM users.

Following must be ensured while specifying the SNMP identifier:

- Only digits must be specified
- It should range between 1 - 65535
- It should be unique for each NMM user

The command `tp_manage_user --info` can be executed to know the SNMP identifiers which are already in use.

SNMP Identifier will be appended to the device type in SNMP traps raised for the NMM users.

In a multi-instance server, multiple devices of same type can run together. This identifier will help in differentiating devices corresponding to a NMM User.

For example, AMS trap generated by AMS running from user whose SNMP trap identifier value "111" is configured.

```
snmptrapd[28475]: 04:24:09 TRAP6.TEXTPASS-GEN-MIB::licGracePeriodActive
TEXTPASS-GEN-MIB::licGracePeriodTimer.0 = Timeticks: (59255100) 6 days, 20:35:51.00
    TEXTPASS-GEN-MIB::deviceType.0 = STRING: "AMS_111" from localhost.localdomain
```

After a new user has been successfully added to the NMM system, a message will be displayed on the console showing the username of the newly created user, its UID, its SNMP trap identifier and the values of the SNMP ports that have been generated for the new user.

```
[root@jura-vm10 bin]# ./tp_manage_user --add_user -p=3500 -snmpid=3501
Addition of new user successful !
USER NAME            : tpuser01
USER UID             : 201
USER SNMP IDENTIFIER : 3501

  =============================================================================
  | PROCESS                                       | PORT NUMBERS              |
  =============================================================================
  | QCLI Server Port                              | 3519                      |
  | External Condition Interface Port             | 3501                      |
  | LGP Query Port                                | 3534                      |
  | RTR SNMP Port                                 | 3502                      |
  | DMF SNMP Port                                 | 3600                      |
  | EC-ABM SNMP Port                              | 3546                      |
  | IIW SNMP Port                                 | 3536                      |
  | LGP SNMP Port                                 | 3531                      |
  | SSI SNMP Port                                 | 3526                      |
  | AMS SNMP Port                                 | 3516                      |
  | FAF SNMP Port                                 | 3511                      |
  | HUB SNMP Port                                 | 3506                      |
  | PBC SNMP Port                                 | 3521                      |
  | MAP-SCR SNMP Port                             | 3541                      |
  | RTR SNMP Trap Port                            | 3504                       |
  | DMF SNMP Trap Port                            | 3601                       |
  | EC-ABM SNMP Trap Port                         | 3547                       |
  | IIW SNMP Trap Port                            | 3537                       |
  | LGP SNMP Trap Port                            | 3532                       |
  | SSI SNMP Trap Port                            | 3527                       |
  | AMS SNMP Trap Port                            | 3517                       |
  | FAF SNMP Trap Port                            | 3512                       |
  | HUB SNMP Trap Port                            | 3507                       |
  | PBC SNMP Trap Port                            | 3522                       |
  | MAP-SCR SNMP Trap Port                        | 3542                       |

  | RTR Watchdog SNMP Trap Port                   | 3505                       |
  | DMF Watchdog SNMP Trap Port                   | 3602                       |
  | EC-ABM Watchdog SNMP Trap Port                | 3548                       |
  | IIW Watchdog SNMP Trap Port                   | 3538                       |
  | LGP Watchdog SNMP Trap Port                   | 3533                       |
  | SSI Watchdog SNMP Trap Port                   | 3528                       |
  | AMS Watchdog SNMP Trap Port                   | 3518                       |
  | FAF Watchdog SNMP Trap Port                   | 3513                       |
  | HUB Watchdog SNMP Trap Port                   | 3508                       |
  | PBC Watchdog SNMP Trap Port                   | 3523                       |
  | MAP-SCR Watchdog SNMP Trap Port               | 3543                       |
  ==============================================================================
```

SNMP ports must be properly configured on MGR while adding new devices on MGR GUI. `tp_manage_user` also creates a `/usr/tpuserxx/.textpass` file for the new user. The content of this file has been mentioned earlier. The newly created user will need an instance specific encrypted license for its functioning. No two different users can use the same license. For more details, please refer to the License section of the NMM components Operator Manual. Also, the semi-configuration file should be placed at the path `/usr/tpuserxx/etc` to start execution of the devices.

3.10.4.2 Deleting a NMM User

An existing NMM user can be deleted by executing the following command as `root` user:

```
/usr/TextPass/bin/tp_manage_user [--delete_user --user_name=<user_name>]
```

Here, `--delete_user` command option is provided to specify that a user is to be deleted.

`--user_name` specifies username of the NMM user to be deleted. User `textpass` cannot be deleted.

Deletion of the user will involve the following:

- `/usr/tpuserxx` and `/var/tpuserxx` folders for the user will be deleted.
- All running processes for the user will be stopped.
- User will be deleted from Server.

On successful deletion of the user, a message will be displayed on the console that the user has been successfully deleted.

3.10.4.3 Display Information For All NMM Users

Information for all existing NMM users can be seen by executing the following command as `root` user:

```
/usr/TextPass/bin/tp_manage_user --info
```

This command will display the following information for all configured NMM users:

- User name
- UID

- Unique SNMP Trap Identifier
- Component-wise SNMP ports

3.10.5 Configuration

For a newly created user, `tpuserxx`, the system's host specific file should be placed at `/usr/tpuserxx/etc` and a correct license (which contains instance specific encryption) should also be installed. For more details about the license, refer to the License section of the NMM component Operator Manual.

3.10.6 Limitations

Following are the limitations for the `tp_manage_user` tool:

- The `tp_manage_user` tool is currently supported on RHEL servers only.
- The multi-instance framework supports functioning for only `RTR`, `HUB`, `PBC`, `AMS`, `LGP`, `FAF`, `IIW` and `SSI` devices.

3.11 tp_restore

The `tp_restore` tool is part of the backup and restore solution for NewNet Mobile Messaging network elements. `tp_restore` will restore from a backup made by `tp_backup`.

See also *tp_backup*.

3.11.1 Synopsis

```
tp_restore
```

3.11.2 Options

Option	Description
`-h, -?, --help`	Print the documentation
`-v, --verbose`	Print status information to `STDOUT`
`-t, --temp`	The location to use as a temporary storage (default: `/var/TextPass/backup`)
`-f, --file`	The filename of the backup file

3.11.3 Usage

The `tp_restore` script will place the files from the backup file back on the system.

It will follow these steps:

1. Create the temporary stage area
2. Unpack the backup files

3. Execute the pre-restore commands from the backup file
4. Move all the files
5. Execute all the post-restore commands
6. Cleanup

During this process detailed information is written to the trace file. This file resides in the same location as the final backup file. The backup file is named: `<hostname><timestamp>.tar.gz` by default. Since the files are system files, only the *root* account can perform a backup.

3.12 tp_set

The `tp_set` tool allows setting the value of an SNMP object.

Important: Caution, any changes to the system configuration made by usage of `tp_set` are not persistent and will be lost after any subsequent restart. It is highly recommended to make the configuration changes in the semi-static configuration files and to apply them by restarting the `textpass` processes.

3.12.1 Synopsis

```
tp_set [<component>] [--device=<host>[:<port>]] <OID>=<value>
```

```
tp_set [<component>] [--device=<host>[:<port>]] <OID> <type> <value>
```

Note:

- To set the special character {` (back tick)} in a string, place the ‘back slash’ before the special character (for example, “\`”).
- To set the special character {! (exclamation point)} in a string, execute the command (set +H) in BASH shell.
- There might be many more special characters that require the above same interpretation.
- Use below format if the component (device type on which configuration needs to be updated) is not installed on the current system:

  ```
  tp_set [<component>] [--device=<host>[:<port>]] <OID> <type> <value>
  ```

- The valid scenarios on the device option with IPv6 address will be:

  ```
  --device=<IPv6>
  --device=[IPv6]:<port>
  ```

  Example:

  ```
  tp_set --device=fe80::20c:29ce:ce00:79ef errClsSriSmMsPurged.0=1
  tp_set --device=[fe80::20c:29ce:ce00:79ef]:11161 errClsSriSmMsPurged.0=1
  ```

- The IPv6 address should be enclosed with '[]' when provided with port.

3.12.2 Options

Option	Description
`--device`	Device

Option	Description
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.12.3 Operands

Operand	Description
`host`	Host name or IP address of the device.
`port`	Port of the device.

3.13 tp_start

The `tp_start` tool starts components. Executing `tp_start` without any options starts all components that are installed and configured.

Note: Executing the `tp_start` tool automatically executes the `tp_config` tool.

3.13.1 Synopsis

```
tp_start [<component>]
```

3.13.2 Options

Option	Description
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.13.3 Operands

`tp_start` has no operands.

3.14 tp_status

The `tp_status` tool provides the operational state and uptime of all installed devices and components.

Note: `tp_status` includes processes that are not SNMP-manageable (such as `tp_trace_receiver`).

3.14.1 Options

`tp_status` has no options.

3.14.2 Operands

`tp_status` has no operands.

3.14.3 Example

The following is a sample output from the `tp_status` tool:

```
PROCESS                 STATE        UPTIME
textpass                Not active   -
tp_ams                  Not active   -
tp_fclient              operating    3 days, 2:01:33.16
tp_qcli                 operating    3 days, 3:01:33.16
tp_mgrd                 operating    3 days, 4:01:33.16
tp_ssi                  operating    3 days, 4:01:33.16
tp_hub*                 adminDisabled 0 days, 3:36:08.78
tp_dmf                  operating    0 days, 0:32:32.56
```

* indicates processes that are running but not configured in the configuration file.

3.15 tp_stop

The `tp_stop` tool stops components. Executing `tp_stop` without any options stops all components that are installed and configured.

3.15.1 Synopsis

```
tp_stop [<component>]
```

3.15.2 Options

Option	Description
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.15.3 Operands

`tp_stop` has no operands.

3.16 tp_system

The `tp_system` tool allows management of the system, including:

- Viewing software and hardware information
- Activating licenses
- Booting the system
- Enabling and disabling subscriptions to the trap service

3.16.1 Synopsis

```
tp_system [<component>] <system>
```

```
tp_system [<component>] --show_licensekey <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --show_licensekey
```

```
tp_system [<component>] --read_licensekey <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --read_licensekey
```

```
tp_system [<component>] --subscribe=<host>:<port> <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --subscribe=<host>:<port>
```

```
tp_system [<component>] --unsubscribe=<host>:<port> <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --unsubscribe=<host>:<port>
```

```
tp_system [<component>] --boot <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --boot
```

```
tp_system [<component>] --traps <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --traps
```

Network Access Function (NAF) Component Only

```
tp_system [<component>] --licensekey=<licensekey> <system>
```

```
tp_system [<component>] --device=<host>[:<port>] --licensekey=<licensekey>
```

3.16.2 Options

Option	Description
`no option`	Provides information about the system: • Software version and hardware type • Licenses that are active • How long the system has been running since its last boot • Alarm stations that are currently subscribed to the trap service
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

Option	Description
--licensekey	Activates a new license key. licensekey specifies the key.
--show_licensekey	Shows the status of the license key.
--read_licensekey	Activates a new license key.
--boot	Soft-boots a system.
--subscribe	Subscribes an alarm station to the trap service. host specifies the host name or IP address of the alarm station. port specifies the UDP port on the alarm station to which the system should send traps.
--unsubscribe	Unsubscribes an alarm station from the trap service. host specifies the host name or IP address of the alarm station. port specifies the UDP port on the alarm station to which the system is sending traps.
--traps	Writes all generic and license-related traps (in a readable format) to the standard output path.

3.16.3 Operands

Operand	Description
<system>	Host name or IP address of the system.

Note: Settings regarding the trap service are stored in volatile memory. Therefore, these settings are lost after a system reset. Only the license is written in non-volatile memory.

3.17 tp_trace_receiver

Trace filtering is a HUB feature that can capture incoming and outgoing UCP, SMPP, and CIMD messages and MXP traffic.

The trace receiver (tp_trace_receiver) tool receives trace data from the trace filters (which are configured in the MGR) and writes the data to a PCAP file. The PCAP file can be read with a tool such as Wireshark.

Generic libraries, such as gen_trace(TCC) and trace_filter(TRF), are provided to configure Mobile Messaging components to send specified trace data to tp_trace_receiver.

tp_trace_receiver should run on a server that has enough disk space to store trace data and that is not used for real-time traffic (that is, it should not run on a server that also runs the AMS, RTR, or HUB).

tp_trace_receiver can be started/stopped by using the tp_start and tp_stoptools with the --trace option.

Refer to the HUB Operator Manual for recommendations for trace receiver usage and instructions for configuring trace filters. Refer to *tp_filter* for information about the command-line tool that enables you to query the current trace filter configuration.

3.17.1 Synopsis

```
tp_trace_receiver [--version] [--fg] [--stderr]
```

3.17.2 Options

Option	Description
--version	Provides the version of tp_trace_receiver.
--fg	Runs tp_trace_receiver in the foreground and disables tp_trace_receiver's watchdog mechanism.
--stderr	Sends all errors and log messages to the standard output.

3.17.3 Configuration File

The tp_trace_receiver configuration should be made in the host-specific configuration file (<hostname>_config.txt).

The following attributes are used to configure tp_trace_receiver:

Parameter	Description
runtraceprocess	tpconfig attribute that specifies if the tp_trace_receiver process should be started. Valid values: • true • false
portnumber	Listener port of tp_trace_receiver. Default: 8200
tracefilename	Prefix of the trace file names. tp_trace_receiver adds the date, time, sequence number, and extension to this prefix. Default: "trc_trace_file"
tracefiledirectory	Directory in which to store trace files. tp_trace_receiver will create the directory if it does not exist. Default: "/var/TextPass/Trace"

Parameter	Description
`maxtracefilesize`	Maximum trace file size (in MB). The maximum configurable size is 4 GB; if a larger value is configured, `tp_trace_receiver` will truncate the file to 4096 MB (for example, if `maxtracefilesize` is set to 5000 MB, `tp_trace_receiver` will truncate the file to be 4096 MB). Default: 10 MB
`idletime`	Amount of time that trace data can be stored in the internal buffer before the data is written to file. This mechanism ensures that the trace file remains up-to-date. Default: 2
`maxnumberoftracefiles`	Maximum number of trace files allowed in the configured directory. `tp_trace_receiver` will automatically remove the oldest file when this number is exceeded. The maximum configurable number of trace files is 5000. Set this parameter to 0 (zero) to disable the mechanism. Default: 100
`sizesocketrcvbuffer`	Socket receive buffer size (in KB), which the system uses to store received trace data. When `tp_trace_receiver` is not scheduled, the kernel fills the socket buffer. The minimum configurable size of the socket buffer is 128 KB; if a lower value is set, the system will use 128 KB. The maximum configurable size of the socket buffer is 2048 KB. `sizesocketrcvbuffer` cannot be set larger than the kernel setting for the maximum UDP socket buffer size. When a larger `sizesocketrcvbuffer` is required, the kernel setting must be changed. Default: 256 kB

The following is an example of a `tp_trace_receiver` configuration section:

```
<tpconfig
  runtextpassprocess="false"
  runtraceprocess="true"
  >

  <trace_receiver
      portnumber="8200"
      tracefiledirectory="/var/TextPass/TraceData"
      tracefilename="trace_data"
      maxtracefilesize="100"
      idletime="2"
      maxnumberoftracefiles="100"
      sizesocketrcvbuffer="1024">
  </trace_receiver>

</tpconfig>
```

3.17.4 Sample Usage

The following command starts the trace receiver (using `tp_start`) with the watchdog enabled and sends all error and log messages to the syslog:

```
tp_start --trace
```

The following command runs `tp_trace_receiver` in the foreground and sends all error and log messages to the standard output:

```
tp_trace_receiver --fg --stderr
```

3.17.5 System Log Messages

`tp_trace_receiver` automatically verifies that received `pcap` frames (network packets) are in-sequence or have not been lost. The tool notifies the user of missing or out-of-sequence frames by writing system log (syslog) messages to the trace file. Possible messages are:

Message	Description
Lost trace message, missing sequence number XXX	Indicates that the trace receiver is missing one single `pcap` frame.
Sequence number reset to 0; client has probably restarted	The trace receiver has unexpectedly received a `pcap` frame with sequence number 0, and has taken this to indicate that the client (HUB) has restarted.
Lost DD trace messages, from seq_nr:XXX to seq_nr:YYY	Indicates that the trace receiver is missing a number of `pcap` frames in a row.
Trace message out of sequence, expected seq_nr XXXX received seq_nr YYYY	Indicates that the trace receiver has received an out-of-sequence `pcap` frame.

3.18 tp_walk

The `tp_walk` tool provides the real-time value of any SNMP attribute.

3.18.1 Synopsis

```
tp_walk [<component>] [--device=<host>[:<port>]] [--verbose] [<OID>]
```

3.18.2 Options

Option	Description
no option	Provides general information about the system: • sysDescr • sysUptime • sysObjectID

Option	Description
`--verbose`	Includes data from the following tables: • appCountryStatsTable • appMobNetworkStatsTable
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.

3.18.3 Operands

Operand	Description
`OID`	Object-identifier used to uniquely identify SNMP attributes.

3.19 tp_walkall

The `tp_walkall` tool provides the real-time value of all SNMP attributes.

3.19.1 Synopsis

```
tp_walkall [--device=<host>] [--verbose] [<component>] [<componentport>=port]
```

3.19.2 Options

Option	Description
no option	Provides the values of all SNMP attributes of all products that are configured to run in the configuration file.
`--device`	Identification of the device.
`--verbose`	Includes data from the following tables: • appCountryStatsTable • appMobNetworkStatsTable
`<component>`	Specifies the Mobile Messaging component to act on. Refer to *Component Options* for the available options.
`<componentport>`	To override the default SNMP walk component port, this specifies another component port where to perform the SNMP walk on. Refer to *Component Options* for the available `<componentport>` options.

3.19.3 Operands

Operand	Description
`host`	Host name or IP address of the device.
`port`	Port of the product.

3.20 trap2email

The `trap2email` tool enables sending notification of SNMP alarms via e-mail to a predefined list of up to 30 recipients. For example:

```
From: "operator@textpass.com" <operator@textpass.com>
To: "support@operator.com" <support@operator.com>,
"support@mbalance.com" <support@mbalance.com>
CC: "manager@operator.com" <manager@operator.com>
Date: Wed, 14 Jan 2009 13:51:13 +0100
Subject: Trap alert

13:50:43 TEXTPASS-AMS-MIB::rtrAvailable from mbalance-
054.asd.mbalance.com
13:50:46 TEXTPASS-HUB-MIB::rtrAvailable from mbalance-
054.asd.mbalance.com
13:50:50 TEXTPASS-HUB-MIB::rtrUnAvailable from mbalance-
054.asd.mbalance.com
```

The `trap2email` tool sends an e-mail after a configurable number of seconds have passed (`interval` parameter) or after a configurable number of traps have occurred (`nummessages` parameter), whichever happens first.

The `trap2email` tool connects to the configured mail server when it starts up. If the tool cannot connect to the mail server, it will not complete start-up and will exit.

During operation, the `trap2email` tool only connects to the mail server to send an e-mail. After sending the e-mail, the tool disconnects.

3.20.1 Synopsis

```
trap2email --version
```

3.20.2 Options

Option	Description
`--version`	Provides the `trap2email` version.

3.20.3 Operands

The `trap2email` tool has no operands.

3.20.4 Configuration File

trap2email is configured in the common or host-specific semi-static (XML) configuration file, using the following parameters:

Parameter	Description
runtrap2emailprocess	tpconfig attribute that specifies if the trap2email process should be started. Valid values: • true • false
snmptrapdport	Local port on which trap2email listens for traps
nummessages	Maximum number of traps to send in an e-mail (default 10, maximum 100)
interval	Maximum number of seconds between sending e-mails (default and maximum 600)
mailserver	Host name of the SMTP server to use to send e-mails (default localhost)
authtype	Type of authentication to use when connecting to the SMTP server: • none—No SMTP authentication will be performed (default) • login—trap2email will attempt a log-in authentication when connecting to the SMTP server
authusername	User name to use for authentication when authtype is *login*
authpassword	Password to use for authentication when authtype is *login*
from	Sender address (default localhost@localhost.com)
format	Format string that trap2email will send to snmptrapd; refer to snmptrapd for possible values (default %02.2h:%02.2j:%02.2k %q %v from %A\n).
to	E-mail addresses to place in the TO field (up to 10)
cc	E-mail addresses to place in the CC field (up to 10)
bcc	E-mail addresses to place in the BCC field (up to 10)

The following is an example of the trap2email configuration:

```
<tpconfig
    ipaddress="10.0.0.79"
    runtextpassprocess="false"
    runtrap2emailprocess="true"
   >
   <trap2email
      snmptrapdport="22222"
      nummessages="10"
      interval="60"
      mailserver="localhost"
      authtype="none"
      authusername="user"
      authpassword="Pass"
      from="operator@textpass.com"
      >
       <to>
           <recipient address="operator@textpass.com"/>
           <recipient address="support@operator.com"/>
       </to>
       <cc>
           <recipient address="manager@operator.com"/>
       </cc>
   </trap2email>

</tpconfig>
```

3.21 trap2sms

The `trap2sms` tool enables sending notification of critical SNMP alarms via SMS to a predefined list of up to 10 recipient MSISDNs. The SMS includes the name of the server originating the alarm. For example:

```
2008/10/12 23:59:26 TRAP6.TEXTPASS-GEN-
MIB::deviceOperationalStateChanged TEXTPASS-GEN-
MIB::deviceOperationalState.0 = INTEGER: operating(2) from server1-ams-01
```

To send the SMSs, the `trap2sms` tool connects to a primary SMSC over UCP; if the connection fails, `trap2sms` connects to a secondary SMSC. If the connection to the secondary SMSC fails, `trap2sms` waits for a customizable number of minutes and attempts to reconnect to the primary SMSC.

If `trap2sms` receives multiple traps while it is sending the notification SMSs, it sends only the most recent trap.

3.21.1 Synopsis

```
trap2sms --version
```

3.21.2 Options

Option	Description
`--version`	Provides the `trap2sms` version.

3.21.3 Operands

The `trap2sms` tool has no operands.

3.21.4 Configuration File

`trap2sms` is configured in the common or host-specific semi-static (XML) configuration file, using the following parameters:

Parameter	Description
runtrap2smsprocess	tpconfig attribute that specifies if the trap2sms process should be started. Valid values: • true • false
snmptrapdport	Local port on which trap2sms listens for traps
originator	Numeric or alphanumeric originator address used to send the SMS
smscretrytime	Number of minutes that trap2sms should wait before attempting to reconnect to the primary SMSC after connecting to both SMSCs has failed
inactivitytime	Maximum number of seconds that an open connection to an SMSC may remain silent
applinfo	SMSC connection parameters: • oadc—Numeric or alphanumeric originator address used to log on. • password—Password to use to log on. • windowsize—Maximum allowed number of pending operations (set to a number greater than or equal to the number of recipients). • timeout—Number of seconds to wait before timing out a log-on request or a submit request.
primarysmsc	Identification of the primary SMSC through which the tool should send the SMS: • host—SMSC host name • port—Connection port
secondarysmsc	Identification of the secondary SMSC through which the tool should send the SMS: • host—SMSC host name • port—Connection port
recipients	List of SMS recipients
recipient address	MSISDN of each recipient

The following is an example of the trap2sms configuration:

```
<tpconfig
  runtextpassprocess="false"
  runtrap2smsprocess="true"
  >

  <trap2sms snmptrapdport="22222" originator="trap2sms" smscretrytime="10"
   inactivitytime="0">
```

```
    <applinfo oadc="1234" password="secret" windowsize="10" timeout="5"/>
    <primarysmsc host="10.0.0.79" port="33331"/>
    <secondarysmsc host="10.0.4.30" port="33333"/>
    <recipients>
      <recipient address="31612345678"/>
      <recipient address="31687654321"/>
      <recipient address="31612348765"/>
    </recipients>
  </trap2sms>

</tpconfig>
```

Appendix A

Component Options

Topics:

A.1 Component Options

This table lists the options that are valid when you specify a `<component>` with the `tp_start`, `tp_stop`, `tp_config`, `tp_set`, `tp_get`, `tp_getnext`, `tp_walk`, `tp_walkall`, and `tp_system` tools.

Option	Abbreviated Option
`--textpass`	`-p`
`--tp_hub`	`-h`
`--tp_naf`	`-n`
`--tp_faf`	`-f`
`--tp_pbc`	`-P`
`--tp_ams`	`-a`
`--tp_emg`	`-e`
`--tp_scr`	`-s`
`--tp_iiw`	`-i`
`--tp_lgp`	`-l`
`--tp_cra`	`-C`
`--tp_bat`	`-b`
`--tp_ssi`	Not applicable
`--tp_fclient` *	`-c`
`--trap2sms` *	`-T`
`--trap2email` *	`-E`
`--trace` *	`-t`
`--qclid` *	`-q`
`--tp_mgrd` **	`--mgrd`
`--tp_dmf`	`-d`
`--xs_rms`	Not applicable
`--xs_mod`	Not applicable
`--xs_mlc`	Not applicable
`--xs_dil`	Not applicable
`--xs_cpy`	Not applicable
`--xs_fwd`	Not applicable

Option	Abbreviated Option
--xs_spa	Not applicable
--xs_tie	Not applicable
--xs_bwl	Not applicable
--xs_biv	Not applicable
--xs_crv	Not applicable
--xs_arp	Not applicable
--xs_sig	Not applicable
--ec_abm	Not applicable
--spf_core	Not applicable
--spf_sms	Not applicable
--spf_abllist ***	Not applicable
--spf_ablclear ***	Not applicable

* Only applies to the tp_start and tp_stop tools.

** In a multi-instance setup, only user textpass can start, stop or fetch status of tp_mgrd process.

*** Available in RHEL only.

A.2 Componentport Options

This table lists the options that are valid when you specify a <componentport> with the tp_walkall tool.

Option
--textpassport
--tp_hubport
--tp_nafport
--tp_fafport
--tp_pbcport
--tp_amsport
--tp_emgport
--tp_scrport
--tp_iiwport
--tp_lgpport

Option
`--tp_craport`
`--tp_batport`
`--tp_ssiport`
`--tp_dmfport`
`--xs_rmsport`
`--xs_modport`
`--xs_mlcport`
`--xs_dilport`
`--xs_cpyport`
`--xs_fwdport`
`--xs_spaport`
`--xs_tieport`
`--xs_bwlport`
`--xs_bivport`
`--xs_crvport`
`--xs_arpport`
`--xs_sigport`
`--ec_abmport`
`--spf_coreport`
`--spf_smsport`

Appendix B

References

Topics:

B.1 References

1. NewNet Mobile Messaging RTR Operator Manual
2. NewNet Mobile Messaging HUB Operator Manual
3. NewNet Mobile Messaging AMS Operator Manual
4. NewNet Mobile Messaging MGR Operator Manual
5. NewNet Mobile Messaging FAF Operator Manual

Glossary

A

AMS

Active Message Store

Provides store-and-forward functionality for SMS messages.

C

CC

Country Code

CIMD

Computer Interface for Message Distribution

Proprietary SMSC protocol developed by Nokia.

CLI

Custom LSMS Interface

Command-line interface

Calling Line Identification

D

DMF

Direct Message Filter

Application component that consumes Intercept files generated by RTR, so it must run with RTR on the same Traffic Element. This component will regularly monitor for new Intercept Files generated by the RTR.

E

EC-ABM

External Condition A and B number Modification component

External condition application that provides a configurable manipulation of A (originator) and B (recipient) numbers.

F

FAF

Firewall Advanced Filter

F

Works in combination with the Firewall to filter messages, modify message content, and alert network operators of increases in SMS-related traffic.

G

GB

Gigabyte — 1,073,741,824 bytes

H

HUB

Works in combination with the Router to manage traffic to and from SMS applications.

I

IIW

IMS InterWorking

Works in combination with the router to provide gateway functionality between IMS domain and SS7 domain.

IP

Internet Protocol

IP specifies the format of packets, also called datagrams, and the addressing scheme. The network layer for the TCP/IP protocol suite widely used on Ethernet networks, defined in STD 5, RFC 791. IP is a connectionless, best-effort packet switching protocol. It provides packet routing, fragmentation and re-assembly through the data link layer.

IPv6

Internet Protocol version 6

L

LGP

Log Processor

Collects and processes data for the Log Viewer to display.

M

M

MB

Megabyte — A unit of computer information storage capacity equal to 1,048, 576 bytes.

MGR

A Web-based interface for managing NewNet Mobile Messaging components. Prior to Suite 6, the Configuration Manager (CM) provided this functionality.

MIB

Management Information Database

MSISDN

Mobile Station International Subscriber Directory Number

The MSISDN is the network specific subscriber number of a mobile communications subscriber. This is normally the phone number that is used to reach the subscriber.

MXP

Message eXchange Protocol

NewNet proprietary protocol used for communication between the Mobile Messaging HUB, RTR, and AMS components.

P

PBC

Prepaid Billing Controller

Performs prepaid charging using the Diameter, CAMEL, or SMPP+ interface.

R

RTR

Router

Routes all types of SMS traffic.

S

SMPP

Short Message Peer-to-Peer Protocol

S

An open, industry standard protocol that provides a flexible data communications interface for transfer of short message data.

SMS

Short Message Service

SMSC

Short Message Service Center

SNMP

Simple Network Management Protocol.

An industry-wide standard protocol used for network management. The SNMP agent maintains data variables that represent aspects of the network. These variables are called managed objects and are stored in a management information base (MIB). The SNMP protocol arranges managed objects into groups.

SS7

Signaling System #7

SSI

Service Subscription Information

The Mobile Messaging SSI can be queried to determine the applicable personalized subscriber services of the originator and recipient of the message.

T

Tools

A collection of command-line tools for managing and troubleshooting NewNet Mobile Messaging components.

trap

A mechanism used in the context of SNMP (Simple Network

T

	Management Protocol) for one-way event notification.

U

UCP	Universal Computer Protocol Protocol used to connect to SMSCs.
UDP	User Datagram Protocol

X

XML	eXtensible Markup Language A version of the Standard Generalized Markup Language (SGML) that allows Web developers to create customized tags for additional functionality.