

# **OpenMind SMSC Gateway User Guide**

Release: 3.4.0

Date: September 16, 2008

---

## Contents

<b>CONTENTS</b>	<b>2</b>
<b>REFERENCES</b>	<b>3</b>
<b>GLOSSARY</b>	<b>3</b>
<b>PREFACE</b>	<b>4</b>
ABOUT THIS GUIDE	4
CONVENTIONS	4
<b>1. OVERVIEW</b>	<b>5</b>
1.1 THE GATEWAY FRAMEWORK	5
1.2 OPENMIND SMSC GATEWAY OVERVIEW	5
1.2.1 <i>Network Details</i>	5
1.2.2 <i>Data Version Support</i>	5
1.2.3 <i>Raw File Names</i>	6
1.2.4 <i>File Format</i>	6
<b>2. ENGINE RULES AND CONFIGURATION</b>	<b>7</b>
2.1 TC1000 AND TC2100	7
2.1.1 <i>Rule Configuration</i>	7
2.1.2 <i>Output Filename</i>	7
2.1.3 <i>Output File Content</i>	8
<b>3. POST PARSER RULES AND CONFIGURATION</b>	<b>9</b>
3.1 COPY_HEADER_N_BODY_COUNTERS	9
3.1.1 <i>Rule Configuration</i>	9
3.1.2 <i>Sample Usage</i>	10
3.2 STRIP_COUNTER_BLOCKNAME	10
3.2.1 <i>Rule Configuration</i>	10
3.2.2 <i>Sample Usage</i>	10

## References

Name	Description
[Gateways Install Note]	This document describes the steps required to install and run a Gateway.
[Gateway Framework User Guide]	This use guide describes in detail the functionality of the Gateway Framework, and the standard suite of tools available.

## Glossary

## Preface

### About this Guide

This guide details the vendor specific information on the OpenMind SMSC Gateway. It contains the following information:

- *Chapter 1. Overview.* This chapter gives a brief description of the OpenMind SMSC Gateway and the raw data format it parses.
- *Chapter 2: Engine Rules and Configuration.* This chapter details the rules and their configuration for parsing vendor specific raw data.
- *Chapter 3: Post Parser Rules and Configuration.* This chapter describes Post Parser rules and their configuration provided with this parser.

### Conventions

The following conventions are used in this guide:

**Fixed width**                      Highlights a block of example code, a configuration entry, or a command line instruction

## 1. Overview

### 1.1 The Gateway Framework

The OpenMind SMSC Gateway uses the Gateway Framework as a container for the execution of its engine and post parser stages. The Gateway Framework and the Gateway are decoupled into two separate installations. The Gateway Framework consists of a library of Perl modules that provide functionality such as:

- a container for the execution of the Engine and Post Parser rules for data transformation
- Intermediate (PIF) and output data (LIF) storage and management
- logging utilities
- cleanup and crash recovery
- statistics gathering

The Gateway plugs into the Gateway Framework and extends this functionality to provide the final Gateway that processes the vendor specific data. More information on the standard Gateway configuration is contained in the [Gateway Framework User Guide].

### 1.2 OpenMind SMSC Gateway Overview

The following is an overview of the Gateway.

#### 1.2.1 Network Details

Traffic Control SMS Router is a high performance, multi--protocol message router that allows operators to consolidate legacy SMSC, First Delivery Router, and Application Router platforms into a single, multi--purpose messaging engine.

#### 1.2.2 Data Version Support

Data versions supported by this vendor gateway are TC1000, and TC2100 OpenMind SMSC raw performance data.

### 1.2.3 Raw File Names

The following files are example of the file naming format,

StatDDRSILO1\_200808151545.txt

StatDDRSILO1\_200808151600.txt

Or

StatDDH1\_200808151430.txt

StatDDH1\_200808151445.txt

The filenames are based on the following convention:

Stat<SMSC\_ID>\_<timestamp>.txt

The SMSC\_ID is in the first part of the name:

Stat**DDRSILO1**\_200808151545.txt

The timestamp is in the second part of the name:

StatDDRSILO1\_**200808151545**.txt

Where,

<SMSC\_ID> is the name of the Network Element.

<timestamp> is in the format - yyyymmddhhmm

For the above file format, a single performance data file will contain all measurement counters.

### 1.2.4 File Format

The files are in ASCII format.

```

OPENMIND-MOS-MIB::stat-cstat-fake-counter.0 = Counter32: 0
OPENMIND-MOS-MIB::stat-cassini-smtpsrc-submits.0 = Counter32: 0
OPENMIND-MOS-MIB::stat-frosti-send-ss-request.0 = Counter32: 22
OPENMIND-MOS-MIB::stat-frosti-send-cs-request.0 = Counter32: 5
OPENMIND-MOS-MIB::stat-frosti-process-q-nack.0 = Counter32: 0

```

Each row in the performance raw file follows the following convention:

<block>::<<counter>.0 = <type>: <value>

Where

<block> is the object block name

<counter> is the name of the object counter

<type> is the counter type. It can have one of two distinct values:

- Counter32: pegged counter type
- Gauge32: intensity counter type

<value> is the performance counter value

## 2. Engine Rules and Configuration

The following are the configurable Engine rules for this Gateway.

Vendor Version	Engine Rule Name
TC1000	OPENMIND_SMS1000_GATEWAY
TC2100	OPENMIND_SMS2100_GATEWAY

### 2.1 TC1000 and TC2100

The TC1000 and TC2100 Engine Rules allow you to process different versions of the OpenMind SMSC raw performance data.

The rules share the same configurable options. The sole difference between the TC1000 and TC2100 rule type is the handling of the timestamp that is extracted from the filename.

- TC1000 treats the timestamp as the START\_DATE and START\_TIME
- TC2100 treats the timestamp as the END\_DATE and END\_TIME

#### 2.1.1 Rule Configuration

The following details the vendor specific configuration rule entries for this Gateway.

- GRANULARITY: This is a mandatory scalar entry. It defines the granularity period of the raw data in seconds. For example, if raw data comes every 15 minutes, the granularity period should be set to 900 (15 minutes x 60 seconds).

```
GRANULARITY => '900',
```

#### 2.1.2 Output Filename

The output filename is generated by the Gateway Engine based on the original input filename. The Gateway Engine will produce initial PIF files with the following naming convention:

```
<original-filename>-#-<datestamp>_<timestamp>-#-I.pif
```

Where

<original-filename> is the name of the original file (with the extension stripped)

<datestamp> is the date extracted from the original filename

<timestamp> is the time extracted from the original filename

Example:

Given the original filename:

StatRoumen\_200808081545.txt

The Gateway Engine will produce the following initial PIF file:

StatRoumen\_200808081545-#-08Aug2008\_15:45-#-I.pif

### 2.1.3 Output File Content

Based on the file format defined in section 1.2.4 above, the raw data comes in the following convention:

`<block>::.0 = <type>: <value>`

The Gateway Engine will produce initial PIF files with the following content:

- Block Name = `<block>`
- Counter Names = `<counter>__tnel_<type>`  
`__tnel_<type>` is appended to each counter name. `<type>` is converted to lower case before the counter name concatenation.

Examples:

`OPENMIND-MOS-MIB::stat-cstat-fake-counter.0 = Counter32: 0`

Block = OPENMIND\_MOS\_MIB

Counter Name = stat\_cstat\_fake\_counter\_\_tnel\_counter32

Value = 0

`OPENMIND-MOS-MIB::stat-stingray-connections.0 = Gauge32: 0`

Block = OPENMIND\_MOS\_MIB

Counter Name = stat\_stingray\_connections\_\_tnel\_gauge32

Value = 0

#### Notes:

- For each counter name, careful attention needs to be paid to the 'stat' prefix and '\_\_tnel\_<type>' postfix that is added to the counter name. This needs to be handled appropriately during the Post Parser phase (i.e. stripping the extra fields to produce the actual counter name).

### 3. Post Parser Rules and Configuration

#### 3.1 COPY\_HEADER\_N\_BODY\_COUNTERS

This post parser rule allows counters to be copied from header to body, and vice versa. The counters that are going to be copied shall be listed out in the specific array.

The non-standard configuration entries with this rule are listed below:

##### 3.1.1 Rule Configuration

- **COPY\_HEADER\_COUNTERS\_2\_BODY**: A mandatory field that consists of an array of header counters. These counters shall be copied to the PIF body.

```
COPY_HEADER_COUNTERS_2_BODY => [ 'END_DATE', 'END_TIME', ],
```

- **COPY\_BODY\_COUNTERS\_2\_HEADER**: A mandatory field that consists of an array of PIF body record counters. These counters shall be copied to the PIF header.

```
COPY_BODY_COUNTERS_2_HEADER => [ 'END_DATE', 'END_TIME' ],
```

- **REDUNDANT\_HEADER\_COUNTERS**: An optional field that consists of an array of header counters. These counters shall be removed.

```
REDUNDANT_HEADER_COUNTERS => [ 'END_DATE', 'END_TIME', ],
```

- **REDUNDANT\_BODY\_COUNTERS**: An optional field that consists of an array of body record counters. These counters shall be removed.

```
REDUNDANT_BODY_COUNTERS => [ 'END_DATE', 'END_TIME', ],
```

A full sample configuration is included below:

```
{
  RULE_TYPE => 'COPY_HEADER_N_BODY_COUNTERS',
  RULE_DESC => 'move counters from body to header for sample PIFs',
  INPUT_FILE_DESCRIPTION => [ '.*', ],
  COPY_HEADER_COUNTERS_2_BODY => [ 'END_DATE', 'END_TIME', 'SMSC_ID' ],
  COPY_BODY_COUNTERS_2_HEADER => [ ],
  REDUNDANT_HEADER_COUNTERS => [ 'END_DATE', 'END_TIME', 'SMSC_ID' ],
  REDUNDANT_BODY_COUNTERS => [ ],
  PRODUCE_PIF => 'True',
  OUTPUT_FORMAT => 0,
}
```

### 3.1.2 Sample Usage

Given the following input file, and with the configuration above:

```
## Parser Intermediate File
##START|HEADER
GRANULARITY|SMSC_ID|END_TIME|DURATION|END_DATE
900|mteltc1|15:30|15|04Sep2008
##END|HEADER
##START|CCSD
msg_received|allow|block_ack|block_nack
2|2|0|0
##END|CCSD
```

The specified 3 header record counters shall be moved to the body as shown below:

```
## Parser Intermediate File
##START|HEADER
GRANULARITY|DURATION
900|15
##END|HEADER
##START|CCSD
msg_received|allow|block_ack|block_nack|SMSC_ID|END_TIME|END_DATE
2|2|0|0|mteltc1|15:30|04Sep2008
##END|CCSD
```

## 3.2 STRIP\_COUNTER\_BLOCKNAME

This post parser rule strips the block name string from all data counters. This is a functional requirement specific to the OpenMind SMSC gateway configuration.

### 3.2.1 Rule Configuration

This post parser rule does not have any non-standard configuration entries.

A full sample configuration is included below:

```
{
  RULE_TYPE           => 'STRIP_COUNTER_BLOCKNAME',
  RULE_DESC           => 'Remove block name from counter names',
  INPUT_FILE_DESCRIPTION => ['.*.'],
  PRODUCE_PIF         => 'True',
  OUTPUT_FORMAT       => 'LIF_Writer',
},
```

### 3.2.2 Sample Usage

Given the following input file, and with the configuration above:

```
## Parser Intermediate File
##START|HEADER
GRANULARITY|SMSC_ID|END_TIME|DURATION|END_DATE
900|mteltc1|15:30|15|04Sep2008
```

```
##END | HEADER
##START | CCSD
ccsd_msg_received|ccsd_allow|ccsd_block_ack|ccsd_block_nack|test_counter
2|2|0|0|0
##END | CCSD
```

The blockname that is prefixed to each counter shall be removed while non-prefixed counters shall remain unaffected as shown below:

```
## Parser Intermediate File
##START | HEADER
GRANULARITY | SMSC_ID | END_TIME | DURATION | END_DATE
900|mteltc1|15:30|15|04Sep2008
##END | HEADER
##START | CCSD
msg_received|allow|block_ack|block_nack|test_counter
2|2|0|0|0
##END | CCSD
```