YAML Configuration Reference

This document provides a complete reference for the `config.yaml` file used by OmniRoam.

Table of Contents

- Configuration Structure
- Global Configuration
  - TAC Configuration
  - Home PLMNs
  - CDR Processor Settings
  - File Paths
  - Exporters
  - InfluxDB Configuration
  - Logging
- Partner Configuration
  - IMSI Prefixes
  - Rate Configuration
  - Batch Information
  - Call Type Levels
- Complete Configuration Example
- Sequence Counters (counters.yaml)

Configuration Structure

The `config.yaml` file has two main sections:

```yaml
config:
  # Global system configuration

partners:
  # Partner-specific configurations
```

Global Configuration

TAC Configuration

Maps Tracking Area Codes (TACs) to serving location information for TAP file generation.

```yaml
config:
  tac_config:
    LocationName:
      tac_list: ['1101', '10000', '10100']
      servingBid: 72473
      servingLocationDescription: 'Example Network USA'
      timezone: 'America/New_York'
```

Parameters:

Parameter	Type	Required	Description
`tac_list`	List of strings	Yes	List of TAC values that match this configuration
`timezone`	String	Yes	IANA timezone identifier for UTC offset calculation
`servingBid`	Integer	No	5-digit Billing ID for TAP output. If omitted, BID is excluded from TAP file
`servingLocationDescription`	String	No	Human-readable location description

Notes:

- Per TAP3.12 specification, `geographicalLocation` is optional
- If a CDR's TAC does not match any entry, the system uses `cdr_processor_tz` as the default timezone
- CDRs without matching TAC will not include BID/location in TAP output

Example with multiple locations:

```yaml
config:
  tac_config:
    Global:
      tac_list: ['1101', '10000', '10100', '10200', '1024']
      servingBid: 72473
      servingLocationDescription: 'Example Network USA'
      timezone: 'America/New_York'
    LabNetwork:
      tac_list: ['100']
      servingBid: 46002
      servingLocationDescription: 'Example Lab'
      timezone: 'America/New_York'
```

Home PLMNs

List of home PLMNs (Public Land Mobile Networks) for the operator.

```yaml
config:
  home_plmns:
    - '313380'
    - '310260'
```

Format: 5 or 6-digit PLMN code (MCC + MNC)

- MCC: Mobile Country Code (3 digits)
- MNC: Mobile Network Code (2 or 3 digits)

CDR Processor Settings

```yaml
config:
  cdr_processor_name: Brewster_TAP_1
  cdr_processor_tz: America/New_York
```

Parameters:

Parameter	Type	Description
`cdr_processor_name`	String	Identifier for this CDR processing instance
`cdr_processor_tz`	String	Default timezone for file timestamps and CDRs without matching TAC

File Paths

Configure input and output paths for CDR processing.

```yaml
config:
  cdr_search_paths:
    - '/var/Athonet_CDRs/'
    - '/var/OmniCore_CDRs/'
  tap_output_path: '/etc/pytap3/OutputFiles'
  tap_human_readable_output_path: '/etc/pytap3/OutputFiles_Human'
  tap_in_path: '/home/nick/Documents/PyTAP/TAP_In/'
```

Parameters:

Parameter	Type	Description
`cdr_search_paths`	List of strings	Directories to scan for incoming CDR files
`tap_output_path`	String	Output directory for binary TAP3 files
`tap_human_readable_output_path`	String	Output directory for human-readable TAP3 files
`tap_in_path`	String	Directory for incoming TAP3 files (for RAP processing)

Roaming Filename Filter

```
config:
  roaming_in_filename: False
```

Parameter:

Parameter	Type	Default	Description
`roaming_in_filename`	Boolean	False	If `True`, only process CDR files with "roaming" in the filename

Raw CDR View Settings

```
config:
  raw_cdr_view_max_age: 7
```

Parameter:

Parameter	Type	Unit	Description
`raw_cdr_view_max_age`	Integer	Days	Maximum age of files viewable in Raw CDRs view in Web UI

Exporters

Configure which exporters are enabled for CDR output.

```
config:
  exporters:
    - CSVExporter
    - Aria
    - virtual_exporter
```

Available Exporters:

- `CSVExporter`: Export CDRs to CSV format
- `Aria`: Export to Aria billing system
- `virtual_exporter`: Virtual/test exporter

InfluxDB Configuration

Configure connection to InfluxDB for metrics and monitoring.

```
config:
  influx_db:
    influxDbUrl: 'http://10.3.0.135:8086'
    influxDbOrg: 'omnitouch'
    influxDbBucket: 'Omnicharge_TAP3'
    influxDbToken: 'your-token-here'
```

Parameters:

Parameter	Type	Description
`influxDbUrl`	String	InfluxDB server URL including protocol and port
`influxDbOrg`	String	InfluxDB organization name
`influxDbBucket`	String	InfluxDB bucket name for storing metrics
`influxDbToken`	String	Authentication token for InfluxDB API access

Security Note: Consider using environment variables or secrets management for the token in production environments.

Logging

```
config:
  log_level: DEBUG
```

Valid Log Levels:

- `DEBUG`: Detailed information for diagnosing problems
- `INFO`: Confirmation that things are working as expected
- `WARNING`: Indication of unexpected events
- `ERROR`: Serious problems that prevent functionality
- `CRITICAL`: Very serious errors that may cause program termination

Partner Configuration

Each roaming partner requires a configuration block under the `partners:` section.

Basic Partner Structure

```yaml
partners:
  PartnerName_Live:
    imsi_prefixes:
      - '311480'
    accessPointNameOI: mnc480.mcc311.gprs
    rates:
      unit_price: 0.000476800
      unit_bytes: 1024
    batch_info:
      sender: USAFU
      recipient: USAVZ
      specificationVersionNumber: 3
      releaseVersionNumber: 12
      accountingInfo:
        localCurrency: 'USD'
        tapCurrency: 'USD'
        roundingAction: 'Simple'
        tapDecimalPlaces: 5
    round_up_to: 1024
    call_type_level:
      qci_1: 20
      qci_2: 22
      default: 20
```

IMSI Prefixes

```yaml
partners:
  VZW_Live:
    imsi_prefixes:
      - '311480'

  VZW_Test:
    imsi_prefixes:
      - '311480616872353'
      - '311480616860347'
```

How IMSI Prefix Matching Works:

OmniRoam uses **longest-match-first logic** to match CDRs to partners:

1. Extract the IMSI from the CDR (e.g., `311480616872353`)
2. Evaluate all partner configurations
3. Find the longest matching prefix
4. Apply that partner's rates and configuration

Example Matching:

- IMSI `311480616872353` → Matches `VZW_Test` (15-digit prefix is longer)
- IMSI `311480123456789` → Matches `VZW_Live` (6-digit prefix)

Use Case - Test SIM Separation:

This allows different handling for test vs. production traffic:

```yaml
partners:
  ATT_Test:
    imsi_prefixes:
      - '310410966551547'  # Specific test SIM (15 digits)
      - '310410966551546'
    rates:
      unit_price: 0.0  # No charge for test traffic
    batch_info:
      sender: USAFU
      recipient: USACGTEST  # Test TADIG code
      fileTypeIndicator: T   # Test file marker

  ATT_Live:
    imsi_prefixes:
      - '310410'  # Production range (6 digits)
    rates:
      unit_price: 0.000476800
    batch_info:
      sender: USAFU
      recipient: USACG  # Production TADIG code
```

Access Point Name OI

```yaml
accessPointNameOI: mnc480.mcc311.gprs
```

Format: `mnc{MNC}.mcc{MCC}.gprs`

- Used in TAP3 files to identify the operator's APN structure

Rate Configuration

```yaml
rates:
  unit_price: 0.000476800
  unit_bytes: 1024
```

Parameters:

Parameter	Type	Description
`unit_price`	Decimal	Price per unit in the configured currency (e.g., USD)
`unit_bytes`	Integer	Number of bytes per billing unit

Rate Calculation Example:

```
Usage: 52,428,800 bytes (50 MB)
Unit Size: 1024 bytes
Units: 52,428,800 ÷ 1024 = 51,200 units
Rate: $0.000476800 per 1KB
Charge: 51,200 × $0.000476800 = $24.41
TAP Units: 24,410 (multiplied by 1000 for TAP3 format)
```

Usage Rounding

```
round_up_to: 1024
```

Parameter:

Parameter	Type	Description
`round_up_to`	Integer	Round total usage up to the nearest N bytes before rating

Example:

- Actual usage: 1500 bytes
- `round_up_to: 1024`
- Rounded usage: 2048 bytes (2 × 1024)

Batch Information

Configuration for TAP3 file batch control information.

```yaml
batch_info:
  sender: USAFU
  recipient: USAVZ
  specificationVersionNumber: 3
  releaseVersionNumber: 12
  fileTypeIndicator: T
  accountingInfo:
    localCurrency: 'USD'
    tapCurrency: 'USD'
    roundingAction: 'Simple'
    tapDecimalPlaces: 5
```

Parameters:

Parameter	Type	Required	Description
`sender`	String	Yes	5-character TADIG code for sending operator
`recipient`	String	Yes	5-character TADIG code for receiving operator
`specificationVersionNumber`	Integer	Yes	TAP specification version (typically 3)
`releaseVersionNumber`	Integer	Yes	TAP release version (e.g., 11, 12)
`fileTypeIndicator`	String	No	Set to `T` for test files. Omit for production files

File Type Indicator:

- If set to `T`: TAP filename prefix is `TD` (Test Data) and file is marked as test
- If omitted: TAP filename prefix is `CD` (Commercial Data)

Accounting Information:

Parameter	Type	Values	Description
`localCurrency`	String	ISO 4217	Currency code for local accounting (e.g., 'USD', 'EUR', 'GBP')
`tapCurrency`	String	ISO 4217	Currency code for TAP file charges
`roundingAction`	String	'Up', 'Down', 'Simple'	How to round calculated charges
`tapDecimalPlaces`	Integer	0-5	Number of decimal places for TAP charge amounts

Rounding Actions:

- `Up`: Always round charges up
- `Down`: Always round charges down
- `Simple`: Standard rounding (0.5 and above rounds up)

Call Type Levels

Map QCI (QoS Class Identifier) values to TAP3 Call Type Levels.

```yaml
call_type_level:
  qci_1: 20
  qci_2: 22
  qci_3: 23
  qci_4: 24
  qci_5: 25
  qci_6: 26
  qci_7: 27
  qci_8: 28
  qci_9: 29
  default: 20
```

Standard QCI Mappings:

QCI	Service Type	Typical Call Type Level
qci_1	Conversational Voice	20 or 21
qci_2	Conversational Video	22
qci_3	Real-time Gaming	23
qci_4	Buffered Streaming	24
qci_5	IMS Signaling	20 or 25
qci_6	Interactive Browsing	26
qci_7	Interactive Gaming	27
qci_8	Background	28
qci_9	Background Low Priority	29
default	Unspecified/Unknown	20

Note: The system uses the `default` value when the CDR's QCI does not match any configured mapping.

Complete Configuration Example

The following complete example shows both test and production configurations:

```yaml
config:
  tac_config:
    Global:
      tac_list: ['1101', '10000', '10100', '10200', '1024']
      servingBid: 72473
      servingLocationDescription: 'Example Network USA'
      timezone: 'America/New_York'
    ExampleLab:
      tac_list: ['100']
      servingBid: 46002
      servingLocationDescription: 'Example Lab'
      timezone: 'America/New_York'

  home_plmns:
    - '313380'

  cdr_processor_name: Brewster_TAP_1
  cdr_processor_tz: America/New_York

  cdr_search_paths:
    - '/var/Athonet_CDRs/'
    - '/var/OmniCore_CDRs/'

  tap_output_path: '/etc/pytap3/OutputFiles'
  tap_human_readable_output_path: '/etc/pytap3/OutputFiles_Human'
  tap_in_path: '/home/user/TAP_In/'

  roaming_in_filename: False
  raw_cdr_view_max_age: 7

  exporters:
    - CSVExporter
    - Aria
    - virtual_exporter

  influx_db:
    influxDbUrl: 'http://10.3.0.135:8086'
    influxDbOrg: 'omnitouch'
    influxDbBucket: 'Omnicharge_TAP3'
    influxDbToken: 'your-secure-token-here'

  log_level: DEBUG
```

```yaml
partners:
  # Test SIM Configuration - Specific IMSIs
  VZW_Test:
    imsi_prefixes:
      - '311480616872353'
      - '311480616860347'
    accessPointNameOI: mnc480.mcc311.gprs
    rates:
      unit_price: 0.000476800
      unit_bytes: 1024
    batch_info:
      sender: USAFU
      recipient: USAVZ
      specificationVersionNumber: 3
      releaseVersionNumber: 12
      fileTypeIndicator: T
      accountingInfo:
        localCurrency: 'USD'
        tapCurrency: 'USD'
        roundingAction: 'Simple'
        tapDecimalPlaces: 5
    round_up_to: 1024
    call_type_level:
      qci_1: 21
      qci_2: 22
      qci_3: 23
      qci_4: 24
      qci_5: 25
      qci_6: 26
      qci_7: 27
      qci_8: 28
      qci_9: 29
      default: 20

  # Production Configuration - IMSI Prefix
  VZW_Live:
    imsi_prefixes:
      - '311480'
    accessPointNameOI: mnc480.mcc311.gprs
    rates:
      unit_price: 0.000476800
      unit_bytes: 1024
    batch_info:
      sender: USAFU
```

```
    recipient: USAVZ
    specificationVersionNumber: 3
    releaseVersionNumber: 12
    accountingInfo:
      localCurrency: 'USD'
      tapCurrency: 'USD'
      roundingAction: 'Simple'
      tapDecimalPlaces: 5
  round_up_to: 1024
  call_type_level:
    qci_1: 20
    qci_2: 22
    qci_3: 23
    qci_4: 24
    qci_5: 20
    qci_6: 26
    qci_7: 27
    qci_8: 28
    qci_9: 29
    default: 20
```

Sequence Counters (counters.yaml)

TAP3 file sequence numbers are tracked separately from `config.yaml` in a `counters.yaml` file. Each TAP3 file name embeds a 5-digit sequence number that must increase monotonically per sender/recipient and file type, so OmniRoam persists the last used value here and increments it every time a file is generated.

The file is a flat map keyed by the 5-character TADIG code, with a separate counter for each file type:

```yaml
AAA00:
  CD: 1     # Commercial Data sequence
  TD: 1     # Test Data sequence
AAA01:
  CD: 11
  TD: 50
```

Key	Description
Top-level key	5-character TADIG code of the partner (recipient)
`CD`	Next sequence number for Commercial Data files
`TD`	Next sequence number for Test Data files

Notes:

- Initialize a new partner with `CD: 1` and `TD: 1`.
- Values auto-increment each time a TAP file is generated for that partner.
- Sequence numbers must not exceed `99999` (the 5-digit field limit); monitor and reset in coordination with the partner if a counter approaches the limit.
- Back up `counters.yaml` alongside `config.yaml` so that sequence continuity is preserved across restores.

Configuration Best Practices

1. IMSI Prefix Organization

Order configurations from most specific to most general:

```yaml
partners:
  # Most specific - test SIMs (15 digits)
  ATT_TestSIM1:
    imsi_prefixes:
      - '310410966551547'

  # Less specific - production range (6 digits)
  ATT_Live:
    imsi_prefixes:
      - '310410'
```

2. Test vs Production Separation

Always separate test and production configurations:

```yaml
partners:
  Partner_Test:
    imsi_prefixes: ['310410966551547']
    batch_info:
      fileTypeIndicator: T  # Mark as test file
      recipient: USACGTEST  # Use test TADIG code

  Partner_Live:
    imsi_prefixes: ['310410']
    # No fileTypeIndicator = production
```

3. TAC Configuration

Create separate TAC configs for different locations:

```yaml
config:
  tac_config:
    MainNetwork:
      tac_list: ['1101', '10000']
      timezone: 'America/New_York'

    LabNetwork:
      tac_list: ['100', '200']
      timezone: 'America/Chicago'
```

4. Security

Protect sensitive configuration:

```
# Set restrictive permissions
chmod 600 config.yaml

# Consider using environment variables for tokens
# influxDbToken: ${INFLUX_TOKEN}
```

5. Backup Configuration Files

Regular backups of configuration:

```
# Backup before changes
cp config.yaml config.yaml.backup
cp counters.yaml counters.yaml.backup
```

6. Version Control

Track configuration changes:

```
git add config.yaml
git commit -m "Added new partner configuration for XYZ"
```

Troubleshooting

CDRs Not Matching Partner

Problem: CDRs are not being assigned to the correct partner.

Solution: Check IMSI prefix matching:

1. Verify IMSI in CDR matches a configured prefix
2. Check for overlapping prefixes (longest match wins)
3. Review logs for IMSI matching errors

TAP Files Have Wrong Sequence Numbers

Problem: TAP file sequence numbers are incorrect.

Solution: Check `counters.yaml`:

```
USAVZ:
  CD: 123  # Commercial Data sequence
  TD: 45   # Test Data sequence
```

Missing Location Information in TAP Files

Problem: TAP files do not include BID or location description.

Solution:

1. Verify CDR's TAC matches a configured `tac_list`
2. Ensure `servingBid` is configured for that TAC
3. Check that TAC configuration includes required fields

Rate Calculation Errors

Problem: Charges do not match expected values.

Solution:

1. Verify `unit_price` and `unit_bytes` are correct
2. Check `roundingAction` setting
3. Review `round_up_to` configuration
4. Verify `tapDecimalPlaces` matches partner requirements

Related Documentation

- README.md - Main documentation
- Sequence Counters (counters.yaml) - TAP3 file sequence counters
- TAP3 Specification - GSMA TAP3.12 standard

OmniRoam - Professional roaming revenue management by Omnitouch.