Référence de l'API HTTP

L'API OmniRoam sert des fichiers TAP, des fichiers RAP, des vues CDR, des totaux, le contrôle d'exportation et des métriques. Ce guide liste les points de terminaison.

Table des matières

- Accès
- Fichiers TAP
- Validation TAP
- Fichiers RAP
- Vues CDR
- Téléversement CDR
- Totaux et surveillance
- Exportation TAP
- Métriques

Accès

L'API OmniRoam écoute sur le port 4000. L'API sert du HTTP simple, sans TLS. Tous les points de terminaison sont sous le chemin `/api`. L'API retourne du JSON sans enveloppe.

Ressource	Chemin
Interface utilisateur Swagger	`GET /api/docs`
Schéma OpenAPI	`GET /api/schema`

Fichiers TAP

Méthode et chemin	Objectif	Paramètres
`GET /api/tap_files_in`	Lister les fichiers TAP entrants.	`tadig` (optionnel)
`GET /api/tap_files_in/:filename`	Retourner un fichier TAP entrant en JSON.	`tadig`, `page`, `per_page`
`GET /api/tap_files_out`	Lister les fichiers TAP sortants.	`tadig` (optionnel)
`GET /api/tap_files_out/:filename`	Retourner un fichier TAP sortant en JSON.	`tadig`, `page`, `per_page`

La réponse du fichier contient l'en-tête et une page d'événements d'appel. La valeur `page` commence à `1`. La valeur `per_page` a une valeur par défaut de `100` et un maximum de `2000`. La réponse retourne `404` lorsque le fichier n'est pas trouvé.

Validation TAP

Méthode et chemin	Objectif	Paramètres
`GET /api/tap_validate_in/:filename`	Valider un fichier TAP entrant.	`tadig` (optionnel)
`GET /api/tap_validate_out/:filename`	Valider un fichier TAP sortant.	`tadig` (optionnel)

La réponse indique si le fichier est valide, le nombre de constatations par gravité, et la liste des constatations. Voir Validation TAP.

Fichiers RAP

Méthode et chemin	Objectif	Paramètres ou corps
`GET /api/rap_files`	Lister les fichiers RAP.	`tadig` (optionnel)
`GET /api/rap_files/:filename`	Retourner un fichier RAP en JSON enrichi.	`tadig` (optionnel)
`POST /api/rap_files/send_alert`	Envoyer une alerte par email pour un fichier RAP.	Corps : `filename` (obligatoire), `tadig`

Voir Traitement RAP.

Vues CDR

Méthode et chemin	Objectif	Paramètres
`GET /api/part_cdrs/:page`	Lister les clés CDR partielles en vol depuis Redis.	Chemin : `page` (entier)
`GET /api/cdrs_from_csv`	Lister les fichiers CDR bruts, ou en télécharger un.	`filename`, `raw_cdr_view_max_age`
`GET /api/config_yaml`	Retourner `config.yaml`. L'API supprime le token InfluxDB.	Aucun
`GET /api/config`	Retourner la section `config` et les compteurs.	Aucun

Téléversement CDR

Méthode et chemin	Objectif	Corps
`POST /api/ui_upload`	Téléverser un fichier CDR CSV dans le répertoire OmniCore.	Multipart `file`. Le nom doit être un nom de fichier nu se terminant par `.csv` — sans séparateurs de chemin. Retourne 400 (rejeté), 409 (déjà détenu), 413 (trop grand).

La réponse retourne `200` en cas de succès, `409` pour un fichier dupliqué, et `500` pour un fichier invalide. Voir Traitement CDR.

Totaux et surveillance

Méthode et chemin	Objectif	Paramètres
`GET /api/monitoring`	Retourner les totaux de surveillance, ou télécharger un fichier correspondant.	`filename` (optionnel)
`GET /api/totals_monthly`	Lister les partenaires avec des totaux mensuels.	Aucun
`GET /api/totals_monthly/:tadig`	Retourner les totaux mensuels pour un TADIG.	`year`, `month`
`GET /api/totals_daily`	Retourner les totaux quotidiens pour tous les TADIGs.	Aucun
`GET /api/totals_daily/:tadig`	Retourner les totaux quotidiens pour un TADIG.	Aucun

Exportation TAP

Méthode et chemin	Objectif	Corps
`POST /api/tap_export`	Mettre en file d'attente une exportation pour un partenaire.	`partner` (obligatoire), `records_per_file`, `max_files`
`GET /api/tap_export`	Lister les travaux d'exportation.	Aucun
`GET /api/tap_export/:job_id`	Retourner un travail d'exportation.	Aucun
`GET /api/export_schedule`	Retourner l'état du planificateur et l'état de la file d'attente.	Aucun
`POST /api/export_schedule`	Mettre en file d'attente une exportation pour chaque partenaire maintenant.	Aucun

Les points de terminaison d'exportation retournent `202` lorsque OmniRoam met le travail en file d'attente. La file d'attente des travaux exécute une exportation à la fois. Voir le Guide des opérations.

Métriques

Méthode et chemin	Objectif
`GET /api/metrics`	Retourner les métriques Prometheus.

La réponse est une exposition de texte Prometheus. Un scrape lit un instantané mis en cache. Voir la Référence des métriques.

- Aperçu

Architecture

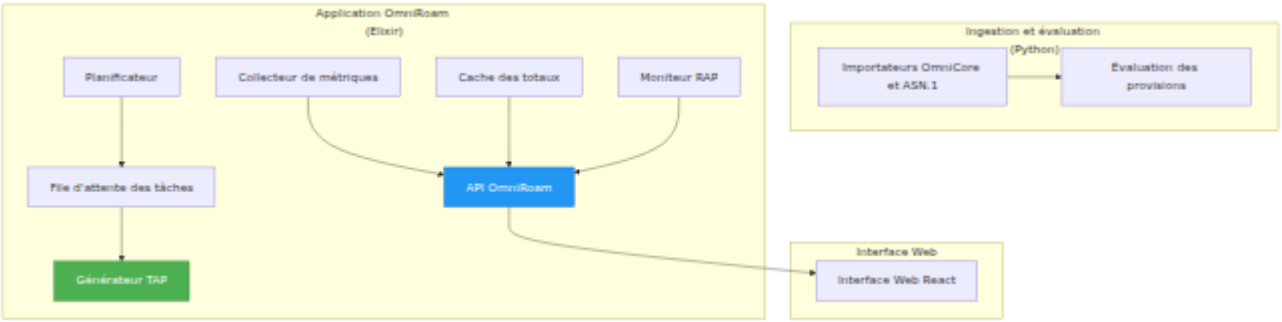
Ce guide décrit les composants d'OmniRoam, le flux de données de bout en bout et les systèmes externes avec lesquels OmniRoam s'intègre.

Table des matières

- Aperçu des composants
- Packages déployables
- Flux de données
  - Ingestion et évaluation
  - Génération de fichiers TAP
  - Transfert de fichiers
  - Retour RAP
- Points d'intégration

Aperçu des composants

OmniRoam se compose de deux parties. Un ensemble d'outils Python ingère et évalue les CDR. Une application Elixir génère des fichiers TAP, traite les fichiers RAP et sert l'API et les métriques.



Ingestion et évaluation (Python)

Ces outils s'exécutent en tant que tâches planifiées. Ils ne fonctionnent pas en tant que services à long terme.

- **Importateur OmniCore** - Lit les fichiers CSV de passerelle de service OmniCore, corrèle les enregistrements partiels en sessions de données complètes et écrit les sessions dans Redis.
- **Importateur ASN.1** - Convertit les fichiers CDR ASN.1 3GPP en XML et les importe.
- **Importateur Redis** - Vidange la file d'attente Redis, applique des filtres d'âge, arrondit l'utilisation et envoie chaque enregistrement au moteur de facturation pour évaluation.
- **Provisionneur d'évaluation** - Charge les plans tarifaires par partenaire dans le moteur de facturation à partir de `config.yaml`.

Application OmniRoam (Elixir)

L'application OmniRoam fonctionne en tant que service systemd. Elle démarre un ensemble de processus supervisés.

Processus	Rôle
Index de fichiers	Met en cache les métadonnées des fichiers TAP et RAP analysés pour un listing rapide.
Cache des totaux	Met en cache les totaux mensuels et quotidiens par partenaire.
Collecteur de métriques	Construit l'instantané des métriques Prometheus en arrière-plan.
File d'attente des tâches	Exécute une exportation TAP à la fois.
Planificateur	Démarre l'exportation quotidienne à l'heure configurée.
API OmniRoam	Sert l'API HTTP sur le port 4000.
Moniteur RAP	Scanne les nouveaux fichiers RAP et envoie des alertes.
Séquence	Réserve les numéros de séquence des fichiers TAP et écrit `counters.yaml`.
Nettoyeur d'espaces	Supprime les espaces indésirables des champs CDR évalués dans la base de données.

Le **Générateur TAP** lit les CDR évalués, associe chaque CDR à un enregistrement TAP, assemble le lot, encode le lot en tant que fichier TAP binaire et marque les CDR comme traités. Voir Génération de fichiers TAP pour plus de détails.

Interface Web

L'interface web est une application monopage React. Elle est servie en tant que contenu statique par nginx. L'interface web appelle l'API OmniRoam sur le port

4000.

Registraire

Le registraire est un connecteur sortant optionnel. Le registraire ouvre un WebSocket vers un système de gestion et transmet les demandes de ce système à l'API OmniRoam locale. Utilisez le registraire pour accéder à l'API depuis derrière un NAT sans port entrant.

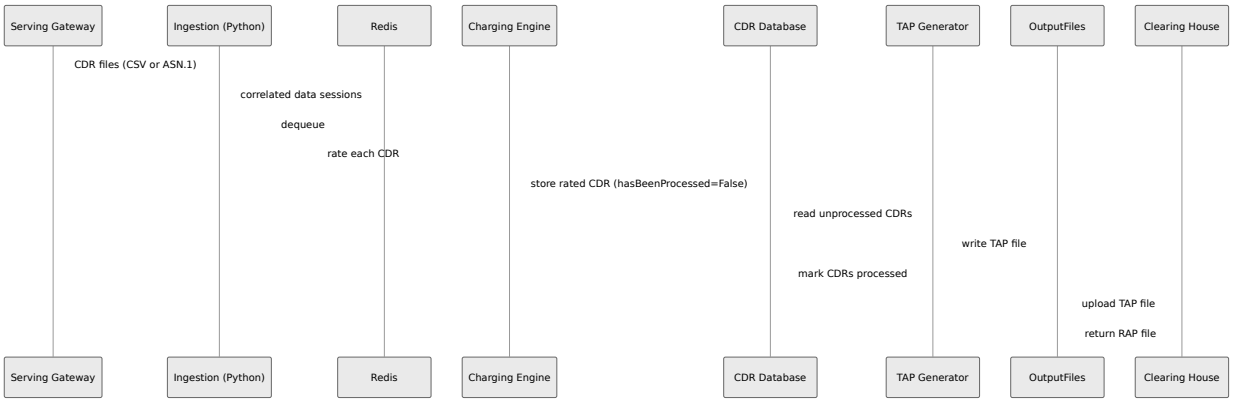
Packages déployables

OmniRoam est livré sous forme d'un ensemble de packages Debian. Le méta-package `omniroam` dépend des packages de composants et verrouille leurs versions ensemble.

Package	Contenu	Runtime
`omniroam`	Méta-package. Dépend des packages de composants.	Aucun.
`omniroam-api`	La version de l'application Elixir.	Service systemd sur le port 4000, utilisateur `omniroam`.
`omniroam-generator`	Les outils d'ingestion et d'évaluation Python plus la configuration.	Aucun service. Exécuté par cron ou manuellement.
`omniroam-ui`	L'interface web statique et le site nginx.	nginx sur le port 80.
`omniroam-registrar`	Le connecteur de registraire sortant.	Service systemd.

Flux de données

Un CDR passe par quatre étapes : ingestion et évaluation, génération de fichiers TAP, transfert de fichiers et retour RAP.



Ingestion et évaluation

Deux sources de CDR alimentent un pipeline.

1. **Passerelles de service OmniCore** envoient des fichiers CDR CSV. Les fichiers arrivent par rsync ou par un téléchargement HTTP vers l'API. OmniRoam écrit les fichiers dans un répertoire local.
2. **Sources CDR ASN.1 3GPP** fournissent des fichiers CDR ASN.1. OmniRoam récupère les fichiers et les convertit en XML.
3. L'importateur corrèle les enregistrements partiels, écrit des sessions de données complètes dans Redis, puis vide Redis dans le moteur de facturation pour évaluation.

Les CDR évalués sont stockés dans la base de données du moteur de facturation (la table CDR). Chaque CDR porte un marqueur `hasBeenProcessed`. Une valeur de `False` signifie que le CDR a encore besoin d'un fichier TAP. Voir Traitement des CDR pour les détails sur l'ingestion et l'évaluation.

Génération de fichiers TAP

Le planificateur ou une demande API met en file d'attente une tâche d'exportation. La file d'attente des tâches exécute une exportation à la fois. Le générateur lit les CDR non traités pour le partenaire, les plus anciens en

premier, et écrit un ou plusieurs fichiers TAP. Le générateur marque ensuite les CDR comme traités et enregistre le numéro de séquence du fichier. Voir Génération de fichiers TAP.

Transfert de fichiers

Lorsque la file d'attente des tâches se vide, OmniRoam exécute la commande de téléchargement. La commande de téléchargement envoie les nouveaux fichiers TAP à la chambre de compensation. Un travail de récupération séparé récupère les fichiers TAP et RAP entrants. Voir le Guide des opérations pour les tâches de transfert.

Retour RAP

Le partenaire valide chaque fichier TAP. Si le partenaire rejette des enregistrements, le partenaire renvoie un fichier RAP. Le travail de récupération place le fichier RAP dans le répertoire entrant. Le moniteur RAP scanne le répertoire, lit chaque nouveau fichier RAP et envoie une alerte par e-mail. Voir Traitement RAP.

Points d'intégration

Système	Protocole	Objectif
Redis	Redis sur localhost	File d'attente des CDR partiels corrélés avant évaluation.
Moteur de facturation	JSON-RPC sur le port 2080	Évalue les CDR et renvoie le coût.
Base de données CDR	Base de données sur le port 3306	Stocke les CDR évalués et les marque comme traités.
InfluxDB	API HTTP	Stocke les analyses d'utilisation et d'exportation.
Chambre de compensation	SCP	Envoie des fichiers TAP et récupère les fichiers TAP et RAP entrants.
Fournisseur d'e-mails	HTTPS	Envoie des e-mails d'alerte RAP.
Prometheus	Scraping HTTP sur `/api/metrics`	Collecte les métriques OmniRoam.

Voir la Référence de l'API HTTP pour les points de terminaison, et la Référence de configuration pour les paramètres de connexion.

- Aperçu

Traitement des CDR

Ce guide décrit comment OmniRoam ingère les Call Detail Records (CDRs), les évalue et les stocke pour la génération de fichiers TAP.

Table des matières

- Sources de CDR
- Pipeline d'ingestion
- Filtres d'âge
- Arrondissement d'utilisation
- Tarification
- Stockage

Sources de CDR

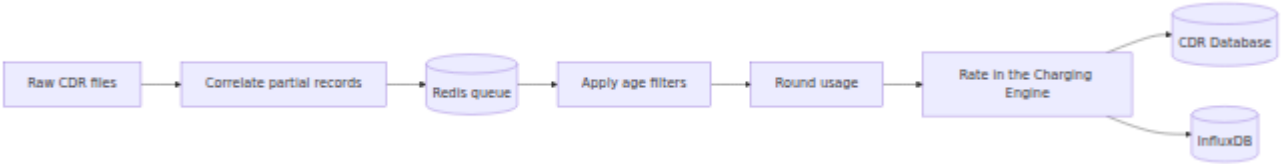
OmniRoam ingère les CDR à partir de deux types de sources.

Source	Format	Transfert	Répertoire de destination
Passerelle de service OmniCore	CSV	rsync, ou un téléchargement HTTP vers l'API	`/var/OmniCore_CDRs`
Source CDR ASN.1 3GPP	ASN.1	rclone	`/var/asn1_cdrs`

Une passerelle de service peut également envoyer un fichier CSV directement à l'API avec une requête `POST /api/ui_upload`. OmniRoam enregistre le fichier téléchargé dans le répertoire OmniCore. Voir la Référence de l'API HTTP.

Pipeline d'ingestion

Les outils d'ingestion s'exécutent en tant que tâches planifiées. Voir le Guide des opérations pour le calendrier.



1. **Corréler.** Une seule session de données peut produire plus d'un CDR partiel. L'importateur joint les enregistrements partiels d'une session en un seul enregistrement complet.
2. **File d'attente.** L'importateur écrit chaque enregistrement complet dans Redis.
3. **Filtrer.** L'importateur Redis vide la file d'attente et supprime les enregistrements qui tombent en dehors de la fenêtre d'âge. L'importateur supprime également les enregistrements ayant une utilisation nulle.
4. **Arrondir.** L'importateur arrondit l'utilisation. Voir Arrondissement d'utilisation.
5. **Tarifer.** L'importateur envoie chaque enregistrement au moteur de tarification pour le prix. Voir Tarification.
6. **Stocker.** Le moteur de tarification stocke le CDR évalué dans la table CDR. L'importateur écrit également un point d'utilisation dans InfluxDB.

Filtres d'âge

L'importateur Redis applique deux filtres d'âge avant de tarifer un enregistrement.

Filtre	Action
Plus vieux que 30 jours	L'importateur supprime l'enregistrement. Un fichier TAP ne peut pas inclure un CDR de plus de 30 jours.
Plus jeune que 24 heures	L'importateur différé l'enregistrement. L'importateur traite l'enregistrement lors d'une exécution ultérieure.

La limite de 30 jours provient des règles GSMA pour les fichiers TAP. Ingérez et tarifez les CDR rapidement, sinon la limite de 30 jours les exclura du règlement.

Arrondissement d'utilisation

OmniRoam arrondit l'utilisation de chaque CDR au multiple supérieur suivant de `round_up_to` octets. La valeur de `round_up_to` est définie par partenaire. Voir la Référence de configuration.

OmniRoam applique deux arrondissements distincts.

1. **Arrondissement par direction.** OmniRoam arrondit le volume entrant et le volume sortant au multiple supérieur suivant, chacun séparément.
2. **Unités facturées.** OmniRoam additionne les volumes entrants et sortants d'origine, puis arrondit ce total au multiple supérieur suivant. Ce total arrondi est l'utilisation facturée que le moteur de tarification évalue.

OmniRoam conserve les volumes d'origine non arrondis pour audit.

Exemple. Un CDR a 500 octets entrants et 700 octets sortants. Le partenaire utilise `round_up_to: 1024`.

- Par direction : entrant arrondit à 1024, sortant arrondit à 1024.
- Unités facturées : le total est de 1200 octets, ce qui arrondit à 2048 octets.

Tarification

OmniRoam évalue chaque CDR avec le moteur de tarification. La tarification se fait lors de l'ingestion, et non lors de la génération du fichier TAP.

1. Le provisionneur de tarifs charge le tarif de chaque partenaire dans le moteur de tarification. Le tarif provient de `unit_price` et `unit_bytes` dans la configuration du partenaire.
2. L'importateur envoie chaque CDR au moteur de tarification avec l'utilisation facturée.
3. Le moteur de tarification renvoie le coût. OmniRoam stocke le coût avec le CDR.

Pour changer le prix d'un partenaire, mettez à jour `unit_price` ou `unit_bytes` dans la Référence de configuration, puis rechargez les tarifs dans le moteur de tarification.

Stockage

OmniRoam stocke un CDR évalué dans la base de données du moteur de tarification. Chaque CDR porte un marqueur `hasBeenProcessed`.

Valeur du marqueur	Signification
`False`	Le CDR est évalué mais pas encore dans un fichier TAP.
Un nom de fichier	Le CDR est dans le fichier TAP nommé.

Le générateur lit les CDR avec `hasBeenProcessed` défini sur `False`, les plus anciens en premier. Voir Génération de fichiers TAP.

OmniRoam écrit également des analyses d'utilisation dans InfluxDB. Voir la Référence des métriques.

- Aperçu

Référence de Configuration

Ce guide documente chaque paramètre dans les fichiers de configuration d'OmniRoam. OmniRoam lit deux fichiers :

- `config.yaml` - La configuration principale. Il a une section `config` globale et une section `partners` par partenaire.
- `counters.yaml` - Les compteurs de séquence de fichiers TAP.

Les outils Python lisent `config.yaml` depuis le répertoire de travail. L'application Elixir lit `config.yaml` depuis le chemin dans la variable d'environnement `OMNIROAM_CONFIG_PATH`, ou depuis `/etc/pytap3/config.yaml`. OmniRoam lit `counters.yaml` depuis le même répertoire que `config.yaml`.

Table des Matières

- Structure de Configuration
- Configuration Globale
  - Configuration TAC
  - PLMNs Domiciliaires
  - Identité du Processeur et Fuseau Horaire
  - Chemins de Fichiers
  - Vue CDR Brute
  - Connexion InfluxDB
  - Journalisation
  - Plan d'Exportation
  - Commande de Complétion d'Exportation
  - Webhooks
  - Enregistrements Par Fichier
- Configuration Partenaire

    - Préfixes IMSI
    - Nom de Point d'Accès
    - Tarifs
    - Arrondi d'Utilisation
    - Informations de Lot
    - Informations Comptables
    - Niveaux de Type d'Appel
- Compteurs de Séquence
- Clés Obsolètes

Structure de Configuration

```
config:
  # Configuration système globale

partners:
  # Configuration par partenaire
```

Configuration Globale

Configuration TAC

`config.tac_config` associe un Code de Zone de Suivi (TAC) à un emplacement de service. OmniRoam utilise l'emplacement de service pour définir l'identifiant de facturation et le fuseau horaire dans chaque enregistrement TAP. Chaque entrée est un modèle nommé avec une liste de TACs.

```
config:
  tac_config:
    ExampleNetwork:
      tac_list: ['1101', '10000', '10100']
      servingBid: 12345
      servingLocationDescription: 'Example Network'
      timezone: 'Etc/UTC'
```

Paramètre	Type	Requis	Par Défaut	Descriptio
`tac_list`	Liste de Chaînes	Oui	-	Les valeurs T qui correspondent ce modèle. OmniRoam compare le T d'un CDR avec cette liste.
`servingBid`	Entier	Non	-	L'identifiant de facturation à chiffres écrit dans l'enregistrem TAP. Omettre pour laisser l champ vide.
`servingLocationDescription`	Chaîne	Non	-	La descriptio de l'emplaceme écrite dans l'enregistrem TAP.
`timezone`	Chaîne	Oui	-	Le nom de fuseau horai IANA. OmniRoam utilise ce fus horaire pour calculer le décalage UT

Paramètre	Type	Requis	Par Défaut	Descriptio
				pour les CDR correspondar

PLMNs Domiciliaires

`config.home_plmns` liste les préfixes PLMN domiciliaires. OmniRoam considère un CDR dont l'IMSI commence par un préfixe PLMN domiciliaire comme un abonné domestique, et non comme un abonné en itinérance. Les outils d'ingestion lisent cette clé.

```yaml
config:
  home_plmns:
    - '001010'
```

Paramètre	Type	Requis	Par Défaut	Description
`home_plmns`	Liste de Chaînes	Oui	-	Les préfixes IMSI (les 6 premiers chiffres) qui identifient un abonné domestique.

Identité du Processeur et Fuseau Horaire

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

Paramètre	Type	Requis	Par Défaut	Description
`cdr_processor_tz`	Chaîne	Oui	-	Le fuseau horaire IANA par défaut. OmniRoam utilise ce fuseau horaire pour les horodatages de fichiers et pour un CDR qui ne correspond à aucun modèle TAC.

Chemins de Fichiers

```yaml
config:
  cdr_search_paths:
    - '/var/asn1_cdrs/'
    - '/var/OmniCore_CDRs/'
  tap_output_path: '/etc/pytap3/OutputFiles'
  tap_human_readable_output_path: '/etc/pytap3/OutputFiles_Human'
  tap_in_path: '/etc/pytap3/TAP_In'
  roaming_in_filename: False
```

Paramètre	Type	Requis	Par Défau
`cdr_search_paths`	Liste de Chaînes	Oui	-
`tap_output_path`	Chaîne	Oui	`/etc/pytap3/Outp`
`tap_human_readable_output_path`	Chaîne	Oui	-
`tap_in_path`	Chaîne	Non	`/etc/pytap3/TAP_`
`roaming_in_filename`	Booléen	Non	`False`

Paramètre	Type	Requis	Par Défaut

Vue CDR Brute

```
config:
  raw_cdr_view_max_age: 7
```

Paramètre	Type	Requis	Par Défaut	Description
`raw_cdr_view_max_age`	Entier	Non	-	L'âge maximum en jours pour les CDR bruts que l'interface web affiche. Une requête peut remplacer cette valeur avec un paramètre de requête.

Connexion InfluxDB

`config.influx_db` définit la connexion InfluxDB pour l'utilisation et l'analyse des exportations. Les quatre valeurs doivent être présentes, sinon OmniRoam

ignore l'écriture de la métrique. Voir la Référence des Métriques pour les mesures.

```yaml
config:
  influx_db:
    influxDbUrl: 'http://influxdb.example.com:8086'
    influxDbOrg: 'example-org'
    influxDbBucket: 'tap3'
    influxDbToken: '<influx-token>'
```

Paramètre	Type	Requis	Par Défaut	Description
`influxDbUrl`	Chaîne	Oui	-	L'URL de base InfluxDB.
`influxDbOrg`	Chaîne	Oui	-	L'organisation InfluxDB.
`influxDbBucket`	Chaîne	Oui	-	Le bucket InfluxDB.
`influxDbToken`	Chaîne	Oui	-	Le token API InfluxDB. L'API supprime cette valeur de la réponse `config_yaml`.

Définissez `influxDbToken` avec votre propre token. Ne commettez pas un vrai token dans le contrôle de version.

Journalisation

```yaml
config:
  log_level: DEBUG
```

Paramètre	Type	Requis	Par Défaut	Description
`log_level`	Chaîne	Oui	-	Le niveau de journalisation pour les outils Python. Utilisez l'un de `DEBUG`, `INFO`, `WARNING`, ou `ERROR`.

Plan d'Exportation

`config.export_schedule` contrôle l'exportation automatique quotidienne. Le planificateur lit cette section à chaque tick, donc un changement s'applique sans redémarrage. Voir le Guide des Opérations pour le comportement du planificateur.

```yaml
config:
  export_schedule:
    enabled: false
    at: '02:00'
    partners: all
```

Paramètre	Type	Requis	Par Défaut	Description
`enabled`	Booléen	Non	`false`	L'interrupteur principal pour l'exportation quotidienne. Réglez sur `true` pour exécuter l'exportation automatiquement.
`at`	Chaîne	Non	-	L'heure d'exécution quotidienne au format `HH:MM`, en UTC.
`partners`	Chaîne ou Liste	Non	`all`	Les partenaires à exporter. Utilisez `all` pour chaque partenaire, ou une liste de noms de partenaires.

Commande de Complétion d'Exportation

`config.on_export_complete` définit une commande qu'OmniRoam exécute lorsque la file d'attente des travaux d'exportation se vide. Utilisez cette commande pour envoyer les nouveaux fichiers TAP à la chambre de compensation.

```yaml
config:
  on_export_complete: '/etc/pytap3/smart_tap_upload.sh'

  # Préféré : invoqué une fois par fichier TAP en attente, avec le
chemin comme argument
  # et TAP_FILE / TAP_FILENAME / TAP_TADIG / TAP_SENT_DIR dans
l'environnement.
  # Sortie 0 et OmniRoam déplace le fichier vers OutputFiles_Sent
; tout autre statut
  # le laisse en attente pour réessayer lors de la prochaine
exécution. Le script ne globule pas, ne déplace pas les fichiers,
  # ou ne suit pas les échecs partiels — donc il fait généralement
dix lignes. Prend la priorité sur
  # on_export_complete, qui reste pour les déploiements qui ne
sont pas encore migrés.
  on_tap_file: '/etc/pytap3/send_one.sh'

  # Entrant : exécute le script de récupération du site selon un
calendrier au lieu de cron.
  # Invoqué avec TAP_IN_DIR et RAP_IN_DIR dans l'environnement ;
sortie 0 signifie que
  # la récupération est terminée (qu'elle ait trouvé quelque chose
ou non). Les exécutions sont sérialisées sur
  # leur propre verrou, les échecs sont comptés, et
tap3_inbound_newest_age_seconds
  # fait une chambre de compensation qui a cessé de répondre à une
alerte plutôt qu'une semaine tranquille.
  # Non défini et rien ne s'exécute.
  on_fetch_files: '/etc/pytap3/pull_tap.sh'
  fetch_interval_minutes: 60
```

Paramètre	Type	Requis	Par Défaut	Description
`on_export_complete`	Chaîne	Non	-	La commande shell à exécuter après la complétion de l'exportation. Laissez vide pour désactiver.

Webhooks

`config.webhooks` définit les URL de webhook entrants Slack. OmniRoam publie un message lorsque l'exportation se termine ou échoue.

```yaml
config:
  webhooks:
    on_complete: '<slack-webhook-url>'
    on_failure: '<slack-webhook-url>'
```

Paramètre	Type	Requis	Par Défaut	Description
`on_complete`	Chaîne	Non	-	L'URL du webhook Slack pour une exportation réussie. Laissez vide pour désactiver.
`on_failure`	Chaîne	Non	-	L'URL du webhook Slack pour une exportation échouée. Laissez vide pour désactiver.

Enregistrements Par Fichier

`config.records_per_tap_file` définit une limite globale sur le nombre de CDRs dans un fichier TAP. Une valeur par partenaire et une valeur par demande remplacent cette valeur globale. Lorsqu'un partenaire a plus de CDRs non traités que la limite, OmniRoam écrit plus d'un fichier TAP.

```yaml
config:
  records_per_tap_file: 140000
```

Paramètre	Type	Requis	Par Défaut	Description
`records_per_tap_file`	Entier	Non	-	Le nombre maximum de CDRs par fichier TAP.

Configuration Partenaire

`partners` contient une entrée par partenaire d'itinérance. Le nom de l'entrée est une étiquette que vous choisissez. Les outils d'ingestion utilisent `imsi_prefixes` pour attribuer un CDR à un partenaire. Le générateur utilise le nom du partenaire pour sélectionner le partenaire à exporter.

```yaml
partners:
  ExamplePartner_Live:
    imsi_prefixes:
      - '001020'
    accessPointNameOI: mnc020.mcc001.gprs
    rates:
      unit_price: 0.000476800
      unit_bytes: 1024
    round_up_to: 1024
    batch_info:
      sender: AAAAA
      recipient: BBBBB
      specificationVersionNumber: 3
      releaseVersionNumber: 12
      accountingInfo:
        localCurrency: 'USD'
        tapCurrency: 'USD'
        roundingAction: 'Simple'
        tapDecimalPlaces: 5
    call_type_level:
      qci_1: 21
      default: 20
```

Préfixes IMSI

Paramètre	Type	Requis	Par Défaut	Description
`imsi_prefixes`	Liste de Chaînes	Oui	-	Les préfixes IMSI qui associent un CDR à ce partenaire. Les outils d'ingestion utilisent la correspondance la plus longue.

Nom de Point d'Accès

Paramètre	Type	Requis	Par Défaut	Description
`accessPointNameOI`	Chaîne	Oui	-	L'Identifiant de l'Opérateur du Nom de Point d'Accès. OmniRoam écrit cette valeur dans chaque enregistrement TAP GPRS.

Tarifs

`rates` définit le prix qu'OmniRoam charge dans le Moteur de Facturation pour ce partenaire. Le Moteur de Facturation évalue chaque CDR selon ce tarif. Voir Traitement des CDR pour le flux de tarification.

Paramètre	Type	Requis	Par Défaut	Description
`unit_price`	Flottant	Oui	-	Le prix pour chaque `unit_bytes` d'utilisation, dans la monnaie locale.
`unit_bytes`	Entier	Oui	-	Le nombre de bytes dans une unité de tarif.

Arrondi d'Utilisation

Paramètre	Type	Requis	Par Défaut	Description
`round_up_to`	Entier	Oui	-	L'unité d'arrondi en bytes. OmniRoam arrondit l'utilisation au multiple supérieur de cette valeur. Voir Traitement des CDR.

Informations de Lot

`batch_info` définit l'identité de chaque fichier TAP pour ce partenaire.

Paramètre	Type	Requis	Par Défaut	Descri
`sender`	Chaîne	Oui	-	Votre code TA caractères. Or écrit cette val `batchControl` et dans le nor
`recipient`	Chaîne	Oui	-	Le code TADIC caractères du OmniRoam ut valeur pour le répertoire de nom du fichie compteur de s
`specificationVersionNumber`	Entier	Oui	`3`	La version de spécification T `3` pour TAP 3.
`releaseVersionNumber`	Entier	Oui	`12`	La version de TAP. Utilisez `1` 3.12.
`fileTypeIndicator`	Chaîne	Non	-	Lorsqu'il est p OmniRoam pr fichier de test Génération de TAP. Omettez pour un fichie

Informations Comptables

`accountingInfo` définit la monnaie et le format monétaire de chaque fichier TAP.

Paramètre	Type	Requis	Par Défaut	Description
`localCurrency`	Chaîne	Oui	-	La monnaie locale sous forme de code ISO 4217.
`tapCurrency`	Chaîne	Oui	-	La monnaie de règlement sous forme de code ISO 4217.
`roundingAction`	Chaîne	Oui	-	Le mode d'arrondi monétaire. Utilisez `Up`, `Down`, ou `Simple`.
`tapDecimalPlaces`	Entier	Oui	-	Le nombre de décimales pour les valeurs monétaires TAP. OmniRoam échelonne les valeurs de charge par 1000, donc définissez cette valeur à `3`. Voir Génération de Fichiers TAP.

Niveaux de Type d'Appel

`call_type_level` associe l'Identifiant de Classe de QoS (QCI) d'un CDR au niveau de type d'appel TAP. OmniRoam recherche la clé `qci_<n>` pour le QCI du CDR. Si la clé est absente, OmniRoam utilise `default`.

```yaml
call_type_level:
  qci_1: 21
  qci_2: 22
  default: 20
```

Paramètre	Type	Requis	Par Défaut	Description
`qci_1` à `qci_9`	Entier	Non	-	Le niveau de type d'appel TAP pour chaque valeur QCI.
`default`	Entier	Oui	-	Le niveau de type d'appel TAP pour un QCI qui n'a pas de clé `qci_<n>`.

Compteurs de Séquence

`counters.yaml` contient les numéros de séquence des fichiers TAP. OmniRoam clé le fichier par le **TADIG du destinataire**, avec un compteur séparé pour chaque type de fichier. Voir Génération de Fichiers TAP pour le comportement du compteur.

```yaml
BBBBB:
  CD: 11
  TD: 50
CCCCC:
  CD: 3
  TD: 2
```

Clé	Description
Clé de niveau supérieur	Le code TADIG du destinataire à 5 caractères.
`CD`	Le prochain numéro de séquence pour un fichier en direct (Données Facturables).
`TD`	Le prochain numéro de séquence pour un fichier de test (Données de Test).

Remarques :

- Initialisez un nouveau partenaire avec `CD: 1` et `TD: 1`.
- OmniRoam incrémente le compteur après avoir écrit un fichier.
- Le numéro de séquence est un champ à 5 chiffres. Réinitialisez le compteur avant qu'il n'atteigne `99999`. Voir le Guide des Opérations.
- Sauvegardez `counters.yaml` avec `config.yaml` afin que la séquence continue correctement après une restauration.

Clés Obsolètes

N'utilisez pas ces clés. OmniRoam ne les lit plus dans le chemin de production.

Clé	Statut
`config.exporters`	Obsolète. L'application actuelle ne lit pas cette clé.
`config.cdr_processor_name`	Obsolète. L'application actuelle ne lit pas cette clé.

- Aperçu

Référence des Métriques

OmniRoam expose deux sources de métriques. L'API OmniRoam expose des métriques Prometheus pour l'état du service. Les outils d'ingestion et d'exportation écrivent des analyses d'utilisation dans InfluxDB.

Table des Matières

- Métriques Prometheus
- Statut du Service
- Exemples de Requêtes
- Mesures InfluxDB

Métriques Prometheus

L'API OmniRoam sert des métriques Prometheus à `GET /api/metrics`. Un scrape lit un instantané mis en cache, donc un scrape ne lance aucune requête de base de données. Le collecteur de métriques rafraîchit l'instantané en arrière-plan, environ une fois par minute.

Toutes les métriques OmniRoam sont des jauges.

Métrique	Étiquettes	Description
`tap3_exporter_up`	-	Le collecteur a un instantané frais. `1` signifie oui.
`tap3_service_up`	`service`	Le service est en cours d'exécution. `1` signifie en marche, `0` signifie hors service.
`tap3_cgrates_up`	-	L'API Charging Engine répond.
`tap3_cgrates_goroutines`	-	Le nombre de routines de travail du Charging Engine.
`tap3_cdr_total`	-	Le nombre total de CDR dans la base de données.
`tap3_files_pending`	-	Le nombre de fichiers en attente de téléchargement.
`tap3_files_sent`	-	Le nombre de fichiers qui ont été envoyés.
`tap3_files_total`	-	Le nombre total de fichiers TAP.
`tap3_files_generated_today`	-	Le nombre de fichiers générés

Métrique	Étiquettes	Description
		aujourd'hui.
`tap3_last_file_generated_timestamp`	-	Le temps Unix de la dernière génération de fichier.
`tap3_last_file_sent_timestamp`	-	Le temps Unix du dernier fichier envoyé.
`tap3_partner_files_pending`	`partner`	Le nombre de fichiers en attente par partenaire.
`tap3_partner_files_sent`	`partner`	Le nombre de fichiers envoyés par partenaire.
`tap3_upload_status`	-	Le dernier statut de téléchargement. `1` est un succès, `0` est un échec, `-1` est inconnu.
`tap3_hours_since_last_file`	-	Le nombre d'heures depuis le dernier fichier TAP.

La métrique `tap3_cdr_total` lit un nombre approximatif de lignes. Le collecteur ne lance pas un comptage complet de la table, car un comptage complet sur la grande table CDR affame le pool de base de données d'exportation.

Statut du Service

La métrique `tap3_service_up` rapporte une série pour chaque dépendance. Le collecteur vérifie chaque service avec le gestionnaire de services système.

Étiquette de Service	Composant
`omniroam-api`	L'API OmniRoam.
`cgrates`	Le Charging Engine (étiquette émise littéralement).
`mysql`	La base de données CDR (étiquette émise littéralement).
`redis-server`	La file d'attente d'ingestion.
`influxdb`	La base de données d'analytique.

Exemples de Requêtes

```
# Fichiers TAP en attente de téléchargement
tap3_files_pending

# Heures depuis le dernier fichier TAP, alerte lorsque élevé
tap3_hours_since_last_file > 26

# Partenaires avec des fichiers en attente
tap3_partner_files_pending > 0

# Une dépendance qui est hors service
tap3_service_up == 0
```

Mesures InfluxDB

Les outils d'ingestion et d'exportation écrivent quatre mesures dans InfluxDB. Configurez la connexion dans la Référence de Configuration.

Mesure	Quand elle est écrite	Étiquettes	Cham
`raw_cdr`	Une fois par CDR, après tarification.	`operator`, `input_file`, `apn`, `cellId`, `imsi`, `tac`, `sGWAddress`, `pGWAddress`	`chargeableUn` `chargedUnits`
`omnicore_cdr_part`	Une fois par CDR partiel pendant l'ingestion d'OmniCore.	`accessPointNameNI`, `hpmln`, `sgw_ip`, `tac`	`dataVolumeIn` `dataVolumeOu`
`file_processing`	Une fois par fichier OmniCore traité.	`individual_filename`	`no_imsi_coun` `sgw_cdr_coun` `pgw_cdr_coun` `charging_id_` `roaming_sgw_` `empty_record` `processing_t`
`tap_cdr`	Une fois par fichier TAP exporté.	`operator`, `filename`	`totalcharge`, `totalconsume` `cdr_count`

La mesure `raw_cdr` a une haute cardinalité, car les étiquettes incluent l'IMSI, l'identifiant de cellule et le nom du fichier d'entrée. Gérez la politique de rétention d'InfluxDB pour contrôler la croissance du stockage. Consultez le Guide des Opérations.

- Aperçu

Guide des opérations

Ce guide décrit les tâches planifiées d'OmniRoam, le transfert de fichiers, l'archivage, les emplacements des journaux et la maintenance régulière.

Table des matières

- Aperçu des tâches planifiées
- Ingestion CDR
- Exportation TAP planifiée
- Transfert de fichiers
- Archivage
- Maintenance du compteur de séquence
- Maintenance du stockage
- Emplacements des journaux

Aperçu des tâches planifiées

OmniRoam fonctionne sur un cycle quotidien. Le tableau montre les principales tâches et leur ordre.

Tâche	Exécute	Objectif
Ingestion CDR	Planifiée par cron	Collecte, corrèle et évalue les CDR.
Exportation TAP	Quotidienne, à l'heure configurée	Génère des fichiers TAP par partenaire.
Téléchargement	Après chaque exportation	Envoie les nouveaux fichiers TAP à la chambre de compensation.
Pull entrant	Quotidien, et un tirage complet hebdomadaire	Récupère les fichiers TAP et RAP entrants.
Moniteur RAP	Toutes les 4 heures	Alerte les opérateurs d'un nouveau fichier RAP.
Archivage	Mensuel	Vidange et purge les anciens CDR de la base de données.

Ingestion CDR

La tâche d'ingestion collecte les CDR des passerelles de service, les corrèle et les évalue. La tâche exécute les outils Python dans l'ordre.

1. Récupérer les fichiers CDR ASN.1 3GPP, et supprimer les fichiers locaux de plus de 60 jours.
2. Convertir les fichiers ASN.1 3GPP en XML, et les importer.
3. Récupérer les fichiers CSV de la passerelle de service OmniCore.
4. Corréler les enregistrements OmniCore, et les écrire dans Redis.
5. Vider Redis, évaluer chaque CDR, et stocker le résultat.

Voir Traitement CDR pour les détails du pipeline et les filtres d'âge.

Exportation TAP planifiée

Le planificateur démarre l'exportation quotidienne. Le planificateur lit `config.export_schedule` à chaque tick, donc un changement s'applique sans redémarrage. Voir la Référence de configuration.

- Le planificateur s'exécute lorsque `enabled` est `true`, que l'heure UTC actuelle atteint `at`, et que l'exportation n'a pas été exécutée aujourd'hui.
- Le planificateur met en file d'attente une tâche d'exportation par partenaire cible.
- La file d'attente des tâches exécute une exportation à la fois. Cette limite protège le pool de base de données partagé.
- Lorsque la file d'attente se vide, OmniRoam exécute la commande dans `on_export_complete`.
- OmniRoam publie un message Slack sur le webhook configuré à l'achèvement ou en cas d'échec.

Le planificateur ne lance pas d'exportation supplémentaire au démarrage. Si le service démarre après l'heure quotidienne, la première exportation s'exécute le lendemain.

Pour exécuter une exportation maintenant, utilisez l'API. Voir la Référence de l'API HTTP.

Transfert de fichiers

OmniRoam déplace des fichiers entre le stockage local et la chambre de compensation.

Téléchargement

OmniRoam exécute la commande de téléchargement après chaque exportation, via `on_export_complete`. La commande de téléchargement envoie les nouveaux fichiers TAP à la chambre de compensation, et déplace chaque fichier envoyé vers le répertoire `OutputFiles_Sent`.

Le script de téléchargement s'exécute en tant qu'utilisateur de transfert. La chambre de compensation rejette la clé de l'utilisateur root. Exécutez le téléchargement en tant qu'utilisateur correct, sinon le transfert échouera.

Pull entrant

La tâche de pull récupère les fichiers entrants de la chambre de compensation dans le répertoire TAP entrant. La tâche de pull copie un fichier à la fois, car le compte de la chambre de compensation ne permet pas de transfert en masse.

Préfixe	Type de fichier
`CD`	Un fichier TAP entrant en direct.
`TD`	Un fichier TAP entrant de test.
`RC`	Un fichier RAP qui rejette une soumission en direct.
`RT`	Un fichier RAP qui rejette une soumission de test.

La tâche de pull exécute un pull incrémental quotidien et un pull complet hebdomadaire. La tâche de pull s'exécute en tant qu'utilisateur de transfert.

Archivage

La tâche d'archivage contrôle la croissance de la base de données CDR. La tâche d'archivage vide les CDR traités qui sont plus anciens que la fenêtre de rétention dans un fichier, puis supprime ces lignes.

Planifiez la tâche d'archivage mensuellement. Confirmez que le fichier de vidage est complet avant de vous fier à la suppression. Gardez la fenêtre de rétention cohérente avec votre politique de purge de base de données.

Maintenance du compteur de séquence

Le numéro de séquence du fichier TAP est un champ de 5 chiffres, avec une plage de `00001` à `99999`. OmniRoam ne réinitialise pas automatiquement le compteur.

1. Surveillez les compteurs `CD` et `TD` pour chaque partenaire dans `counters.yaml`.
2. Avant qu'un compteur n'atteigne `99999`, convenez d'une réinitialisation avec le partenaire.
3. Réinitialisez le compteur à `1` au moment convenu.
4. Sauvegardez `counters.yaml` avec `config.yaml`.

Voir la Référence de configuration.

Maintenance du stockage

Surveillez ces magasins pour la croissance.

Magasin	Action
`OutputFiles`	Confirmez que le téléchargement déplace chaque fichier envoyé vers `OutputFiles_Sent`. Un nombre en attente croissant signifie que le téléchargement a échoué. Surveillez `tap3_files_pending` et `tap3_hours_since_last_file`.
InfluxDB	Définissez une politique de rétention. La mesure `raw_cdr` a une haute cardinalité.
Base de données CDR	Exécutez la tâche d'archivage selon le calendrier.
Fichiers journaux	Les outils Python écrivent un nouveau fichier journal horodaté par exécution dans `/tmp`. Supprimez les anciens fichiers journaux.

Voir la Référence des métriques pour les métriques de surveillance.

Emplacements des journaux

Les outils Python écrivent un fichier journal horodaté par exécution.

Modèle de fichier	Source
`/tmp/import_CDR_Logger_Marben_*.log`	Les importateurs ASN.1 et Redis.
`/tmp/import_CDR_Logger_OmniCore_*.log`	L'importateur OmniCore.
`/tmp/tap3_export_*.log`	L'outil d'exportation hérité.
`/tmp/convert_cdr_logger_*.log`	Le convertisseur ASN.1 3GPP en XML.
`/tmp/rate_creator_Logger_*.log`	Le provisionneur de tarifs.

L'application OmniRoam journalise via le gestionnaire de services système. Le script de téléchargement écrit son statut dans un fichier journal que le collecteur de métriques lit.

- Aperçu

Traitement des RAP

Ce guide décrit les fichiers de la Procédure de Compte Retourné (RAP) que les partenaires retournent, et comment OmniRoam alerte les opérateurs à leur sujet.

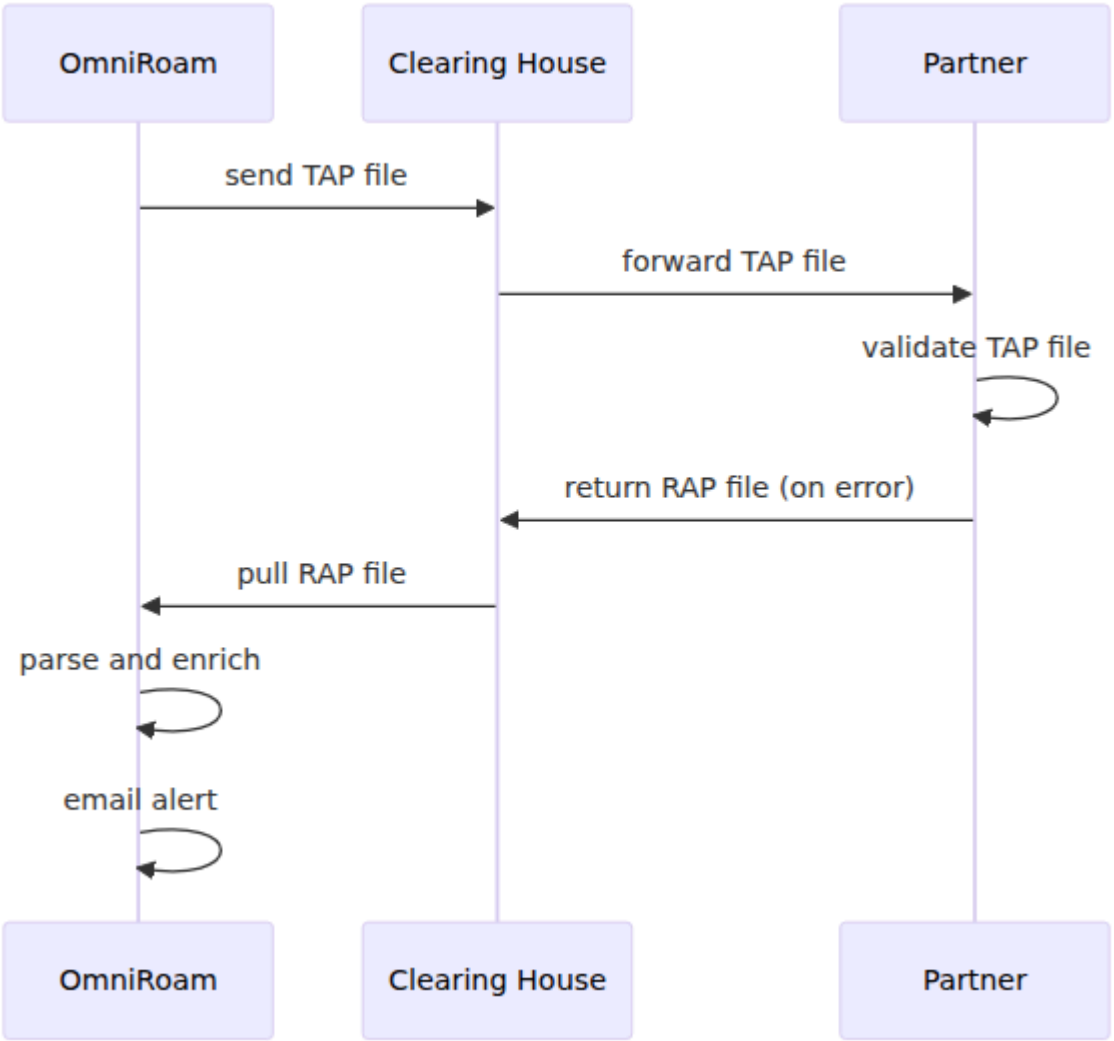
Table des Matières

- Qu'est-ce qu'un fichier RAP
- Nommage des fichiers RAP
- Types de retour et gravité
- Codes d'erreur
- Surveillance et alertes RAP
- Lecture d'un fichier RAP

Qu'est-ce qu'un fichier RAP

Un partenaire valide chaque fichier TAP que OmniRoam envoie. Si le partenaire trouve des erreurs, le partenaire retourne un fichier RAP. Le fichier RAP rapporte les erreurs dans un fichier TAP spécifique. Le format RAP suit GSMA TD.32.

Un fichier RAP est le messager. Les codes d'erreur décrivent les défauts dans le fichier TAP original. OmniRoam lit le fichier RAP, ajoute une description à chaque erreur, et alerte les opérateurs.



Nommage des fichiers RAP

Un nom de fichier RAP a une longueur fixe de 17 caractères. Le nom reflète le nom du fichier TAP, mais utilise un préfixe RAP.

Préfixe	Signification
`RC`	Un retour contre un fichier TAP en direct (Données Facturables).
`RT`	Un retour contre un fichier TAP de test (Données de Test).

Le reste du nom contient l'expéditeur de 5 caractères, le destinataire de 5 caractères, et le numéro de séquence de 5 chiffres. OmniRoam ne lit que les fichiers qui commencent par `RC` ou `RT` comme fichiers RAP.

Types de retour et gravité

Un fichier RAP contient un ou plusieurs retours. Chaque retour a un type et une gravité.

Type de retour	Gravité	Effet
Retour fatal	Critique	Le partenaire rejette l'ensemble du fichier TAP.
Retour sévère	Avertissement	Le partenaire rejette les enregistrements individuels, mais accepte le reste du fichier.
Retour manquant	Avertissement	Le partenaire signale un écart dans la séquence de fichiers.
Retour d'arrêt	Information	Le partenaire signale qu'il n'a reçu aucun fichier TAP pendant 7 jours ou plus.

Traitez d'abord un retour fatal. Un retour fatal signifie que le partenaire a rejeté l'ensemble du fichier, et vous devez corriger le défaut et renvoyer.

Codes d'erreur

OmniRoam ajoute une description à chaque code d'erreur dans un fichier RAP. Les codes proviennent de GSMA TD.57. Les identifiants de chemin proviennent de TD.57 pour le fichier TAP original, et de TD.32 pour le fichier RAP.

Le tableau ci-dessous montre les codes d'erreur courants. Voir Validation TAP pour les catégories.

Code	Description
20	Un élément de données requis est manquant.
50	Une erreur d'encodage RAP dans la structure ASN.1.
107	Une structure invalide.
200	Les données ne sont pas conformes à l'accord d'itinérance.
250 à 257	Un événement d'appel en double.
260	Une erreur d'âge d'appel.
263	Un format de timestamp invalide.

Surveillance et alertes RAP

Le moniteur RAP est un processus en arrière-plan dans l'application OmniRoam. Le moniteur surveille les répertoires entrants et alerte les opérateurs d'un nouveau fichier RAP.

Propriété	Valeur
Intervalle de scan	Toutes les 4 heures.
Répertoires	Le répertoire TAP entrant et le répertoire RAP. Un fichier RAP arrive avec des fichiers TAP entrants.
Méthode d'alerte	Email via le fournisseur d'email configuré.
Dé-duplication	OmniRoam enregistre chaque fichier alerté. OmniRoam n'alerte pas sur le même fichier deux fois.

Une alerte par email nomme le type de retour, le nombre d'erreurs, et le nom du fichier. L'alerte inclut également les informations de lot et les codes d'erreur.

Définissez les destinataires des alertes et les identifiants du fournisseur d'email dans la configuration d'exécution de l'application. Ne stockez pas les identifiants dans `config.yaml`.

Pour envoyer une alerte de test pour un fichier connu, utilisez `POST /api/rap_files/send_alert`. Voir la Référence API HTTP.

Lecture d'un fichier RAP

L'API OmniRoam et l'interface web lisent les fichiers RAP. Une requête à `GET /api/rap_files/:filename` retourne le contenu RAP analysé et enrichi au format JSON.

OmniRoam extrait ces champs d'un fichier RAP :

Champ	Description
`sender`	Le code TADIG du réseau qui a envoyé le fichier RAP.
`recipient`	Le code TADIG du réseau qui reçoit le fichier RAP.
`roamingPartner`	Le code TADIG du partenaire d'itinérance.
`rapFileSequenceNumber`	Le numéro de séquence du fichier RAP.
`rapFileCreationTimeStamp`	L'heure de création du fichier RAP.
`tapCurrency`	La devise de règlement du fichier TAP original.
`tapDecimalPlaces`	Les décimales du fichier TAP original.
`returnDetailsCount`	Le nombre de retours dans le fichier RAP.
`totalSevereReturnValue`	La valeur totale des retours sévères.

- Aperçu

Génération de Fichiers TAP

Ce guide décrit comment OmniRoam construit des fichiers TAP, comment il les nomme et comment il gère les compteurs de séquence.

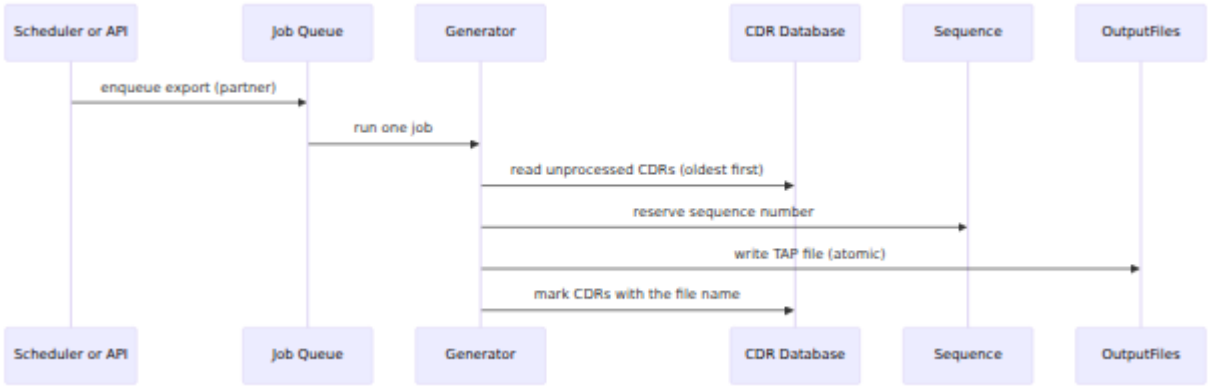
Table des Matières

- Aperçu
- Comment Fonctionne une Exécution de Génération
- Nom des Fichiers
- Fichiers de Test et Fichiers en Direct
- Compteurs de Séquence
- Structure du Fichier TAP
- Valeurs de Charge
- Fichiers NOTIFY
- Déclenchement d'une Exportation

Aperçu

Un fichier TAP est un fichier binaire qui rapporte l'utilisation de l'itinérance facturée à un partenaire. OmniRoam encode chaque fichier TAP en ASN.1 dans le format BER, selon la grammaire TAP 3.12. Le format suit GSMA TD.57.

Comment Fonctionne une Exécution de Génération



1. Le planificateur ou une demande API met en file d'attente un travail d'exportation pour un partenaire.
2. La file d'attente des travaux exécute un travail à la fois. Cette limite protège le pool de connexions de base de données partagé.
3. Le générateur lit les CDR non traités pour le partenaire, du plus ancien au plus récent.
4. Le générateur divise les CDR en fichiers si le nombre dépasse la limite d'enregistrements par fichier.
5. Le générateur réserve un numéro de séquence, encode le fichier et l'écrit de manière atomique.
6. Le générateur marque chaque CDR dans la base de données avec le nom du fichier.

Le générateur écrit chaque fichier TAP dans un sous-répertoire nommé pour le TADIG destinataire, à l'intérieur de `tap_output_path`. Le générateur écrit également une copie JSON lisible par l'homme dans `tap_human_readable_output_path`.

Nom des Fichiers

Un nom de fichier TAP a une longueur fixe de 17 caractères.

```
CD AAAAA BBBBB 00011
```

Partie	Longueur	Description
Préfixe	2	`CD` pour un fichier en direct, ou `TD` pour un fichier de test.
Expéditeur	5	Votre code TADIG, provenant de `batch_info.sender`.
Destinataire	5	Le code TADIG du partenaire, provenant de `batch_info.recipient`.
Séquence	5	Le numéro de séquence avec des zéros en tête.

Le nom `CDAAAAABBBBB00011` signifie un fichier en direct, de `AAAAA`, à `BBBBB`, séquence `11`.

Fichiers de Test et Fichiers en Direct

La clé `fileTypeIndicator` dans `batch_info` sélectionne le type de fichier.

- Si `batch_info` ne contient pas `fileTypeIndicator`, OmniRoam produit un fichier **en direct**. Le nom utilise le préfixe `CD` et le compteur de séquence `CD`.
- Si `batch_info` contient `fileTypeIndicator` avec la valeur `T`, OmniRoam produit un fichier **de test**. Le nom utilise le préfixe `TD` et le compteur de séquence `TD`.

Définissez `fileTypeIndicator` sur `T` pendant que vous intégrez un nouveau partenaire. Supprimez la clé pour passer le partenaire à un règlement en direct. Utilisez uniquement la valeur `T`. Une valeur différente peut produire un nom `TD` avec un numéro de séquence `CD`, ce qui casse l'alignement du compteur.

Compteurs de Séquence

OmniRoam suit un numéro de séquence pour chaque TADIG destinataire et chaque type de fichier. OmniRoam stocke les compteurs dans `counters.yaml`. Voir la Référence de Configuration.

- **CD** compte les fichiers en direct (Chargeable Data).
- **TD** compte les fichiers de test (Test Data).

Le numéro de séquence augmente de un pour chaque fichier. Le nombre n'augmente pas pour une retransmission. Les séquences TAP et RAP sont indépendantes.

OmniRoam réserve un numéro de séquence à travers un seul processus et écrit `counters.yaml` de manière atomique. Cette méthode empêche deux exportations d'utiliser le même numéro. Une écriture échouée garde le fichier de compteur précédent intact.

Le numéro de séquence est un champ de 5 chiffres avec une plage de `00001` à `99999`. OmniRoam ne réinitialise pas le compteur automatiquement. Voir le Guide des Opérations pour la procédure de réinitialisation.

Structure du Fichier TAP

Un fichier TAP est une structure `transferBatch`. Le lot contient des informations de contrôle et une liste d'enregistrements d'événements d'appel. OmniRoam produit uniquement des enregistrements GPRS (données).

Section	Contenu
`batchControlInfo`	L'expéditeur, le destinataire, le numéro de séquence du fichier, les horodatages du fichier et les numéros de version.
`accountingInfo`	La monnaie locale, la monnaie de règlement, les décimales et les informations sur le taux de change.
`networkInfo`	Le tableau des décalages UTC et la liste des entités d'enregistrement (la passerelle de paquets et la passerelle de service).
`callEventDetails`	La liste des enregistrements d'appels GPRS.
`auditControlInfo`	Les horodatages d'appel les plus anciens et les plus récents, le total des charges et le nombre d'enregistrements.

Chaque enregistrement d'appel GPRS contient les champs ci-dessous.

Champ	Description
Abonné facturable	L'IMSI, et le MSISDN lorsqu'il est présent.
Adresse PDP	L'adresse IP de l'abonné. OmniRoam préfère IPv6.
Nom du point d'accès	L'identifiant du réseau et l'identifiant de l'opérateur de l'APN.
Horodatage de début de l'événement d'appel	L'heure de début de facturation, avec le décalage UTC pour le lieu de service.
Durée totale de l'événement d'appel	La durée de la session en secondes.
ID de facturation	L'identifiant de facturation de la session.
Informations de localisation	Les entités d'enregistrement, la zone de localisation, l'identifiant de cellule et l'identifiant de facturation de service optionnel.
Volumes de données	Les comptes de bytes entrants et sortants.
Informations de charge	La charge, les unités facturables, les unités facturées et les niveaux de type d'appel.

Valeurs de Charge

Les fichiers TAP stockent l'argent sous forme d'entier. Le lecteur divise l'entier par `10^tapDecimalPlaces` pour obtenir la valeur décimale.

OmniRoam multiplie chaque charge par 1000 pour produire l'entier. Cette échelle correspond à `tapDecimalPlaces` de `3`. Définissez `tapDecimalPlaces` sur `3` pour chaque partenaire. Une valeur différente produit un entier de charge qui ne correspond pas aux décimales déclarées.

Le paramètre `roundingAction` contrôle comment OmniRoam arrondit la charge mise à l'échelle :

- `Up` arrondit la valeur vers le haut.
- `Down` arrondit la valeur vers le bas.
- `Simple` arrondit la valeur à l'entier le plus proche.

Fichiers NOTIFY

Un partenaire s'attend à un fichier même en l'absence d'utilisation. Lorsqu'un partenaire n'a pas de CDR non traités, OmniRoam écrit un fichier NOTIFY. Un fichier NOTIFY utilise le même format de nom qu'un fichier TAP, mais il contient un enregistrement de `notification` au lieu d'un lot d'événements d'appel.

Déclenchement d'une Exportation

Vous pouvez commencer une exportation de trois manières.

Méthode	Action
Planificateur	OmniRoam exécute l'exportation automatiquement à l'heure configurée. Voir le Guide des Opérations.
API	Envoyez `POST /api/tap_export` pour un partenaire, ou `POST /api/export_schedule` pour chaque partenaire. Voir la Référence de l'API HTTP.
Ligne de commande	Exécutez l'outil d'exportation hérité avec le nom du partenaire. Utilisez cette méthode uniquement comme solution de secours.

- Aperçu

Dépannage

Ce guide répertorie les problèmes courants et leurs résolutions. Chaque entrée donne les symptômes, les causes possibles et les étapes pour résoudre le problème.

Table des matières

- Les CDR n'atteignent pas un partenaire
- Les fichiers TAP ne sont pas téléchargés
- Mauvais numéros de séquence
- Un partenaire renvoie un fichier RAP
- Emplacement manquant dans les fichiers TAP
- Mauvaises valeurs de charge
- Pas de métriques ni d'analytique

Les CDR n'atteignent pas un partenaire

Symptômes : Un partenaire signale une utilisation manquante. Les CDR sont dans la base de données mais pas dans un fichier TAP.

Causes possibles :

- L'IMSI du CDR ne correspond à aucun préfixe de partenaire.
- Le CDR a plus de 30 jours, donc l'ingestion l'a supprimé.
- L'exportation n'a pas été exécutée pour le partenaire.

Résolution :

1. Confirmez que l'IMSI du CDR correspond à un préfixe dans `imsi_prefixes` pour le partenaire.

2. Confirmez que l'âge du CDR est dans les 30 jours. Ingérez et évaluez les CDR rapidement. Voir Traitement des CDR.
3. Confirmez que l'exportation a été exécutée. Vérifiez les travaux d'exportation via la Référence de l'API HTTP.

Les fichiers TAP ne sont pas téléchargés

Symptômes : Le nombre de fichiers en attente augmente. La métrique `tap3_files_pending` est élevée.

Causes possibles :

- La commande de téléchargement dans `on_export_complete` est vide ou incorrecte.
- Le téléchargement s'exécute en tant que mauvais utilisateur, donc la chambre de compensation rejette la clé.
- La chambre de compensation n'est pas accessible.

Résolution :

1. Confirmez que la commande `on_export_complete` est définie. Voir la Référence de configuration.
2. Confirmez que le téléchargement s'exécute en tant qu'utilisateur de transfert, pas root.
3. Vérifiez le journal de téléchargement et la métrique `tap3_upload_status`.

Mauvais numéros de séquence

Symptômes : Un partenaire signale un écart de séquence ou un numéro de séquence dupliqué.

Causes possibles :

- Le fichier `counters.yaml` a été restauré à partir d'une ancienne sauvegarde.

- Un compteur a atteint `99999`.
- La valeur de `fileTypeIndicator` est définie sur une valeur autre que `T`.

Résolution :

1. Vérifiez les compteurs `CD` et `TD` pour le TADIG destinataire dans `counters.yaml`.
2. Si un compteur est proche de `99999`, suivez la procédure de réinitialisation dans le Guide des opérations.
3. Définissez `fileTypeIndicator` sur `T` pour un fichier de test, ou retirez la clé pour un fichier en direct. Voir Génération de fichiers TAP.

Un partenaire renvoie un fichier RAP

Symptômes : Le moniteur RAP envoie une alerte par e-mail. Un fichier RAP est dans le répertoire entrant.

Causes possibles :

- Le fichier TAP a échoué à la validation du partenaire.

Résolution :

1. Lisez le type de retour. Traitez d'abord un retour fatal, car le partenaire a rejeté l'ensemble du fichier.
2. Lisez les codes d'erreur dans l'alerte ou via `GET /api/rap_files/:filename`.
3. Corrigez la faute et renvoyez le fichier. Voir Traitement des RAP.

Emplacement manquant dans les fichiers TAP

Symptômes : Un fichier TAP n'a pas d'identifiant de facturation de service ou de description d'emplacement.

Causes possibles :

- L'IMSI du CDR ne correspond à aucun modèle `tac_config`.
- Le modèle n'a pas de `servingBid`.

Résolution :

1. Confirmez que l'IMSI du CDR est dans une `tac_list`. Voir la Référence de configuration.
2. Confirmez que le modèle a un `servingBid` et une `servingLocationDescription`.

Mauvaises valeurs de charge

Symptômes : Les valeurs de charge ne correspondent pas au montant de règlement attendu.

Causes possibles :

- La valeur de `unit_price` ou `unit_bytes` est incorrecte.
- La valeur de `tapDecimalPlaces` n'est pas `3`.
- La valeur de `roundingAction` est incorrecte.

Résolution :

1. Confirmez les valeurs de `unit_price` et `unit_bytes`. Voir la Référence de configuration.
2. Définissez `tapDecimalPlaces` sur `3`. Voir Génération de fichiers TAP.
3. Confirmez la valeur de `roundingAction`.
4. Rechargez les tarifs dans le moteur de facturation après tout changement de tarif.

Pas de métriques ni d'analytique

Symptômes : Les analyses InfluxDB sont vides. Une métrique Prometheus est absente.

Causes possibles :

- La section `influx_db` est incomplète.
- Un service dépendant est hors service.

Résolution :

1. Confirmez que toutes les quatre valeurs `influx_db` sont définies. OmniRoam ignore l'écriture si une valeur est manquante. Voir la Référence de configuration.
2. Vérifiez la métrique `tap3_service_up` pour un service qui est hors service. Voir la Référence des métriques.

- Aperçu

Validation TAP

OmniRoam valide les fichiers TAP selon les règles vérifiables dans GSMA TD.57 et GSMA TD.32. Vous pouvez valider un fichier TAP entrant ou un fichier TAP sortant via l'API.

Table des matières

- Comment valider un fichier
- Vérifications de validation
- Gravité des erreurs
- Catégories de codes d'erreur
- Limites de la validation

Comment valider un fichier

Utilisez l'API pour valider un fichier TAP.

Point de terminaison	But
`GET /api/tap_validate_in/:filename`	Valider un fichier TAP entrant.
`GET /api/tap_validate_out/:filename`	Valider un fichier TAP sortant.

La réponse indique si le fichier est valide et compte les résultats par gravité. Chaque résultat contient le code d'erreur, la gravité, l'emplacement et un message. Voir la Référence API HTTP.

Vérifications de validation

OmniRoam effectue quatre groupes de vérifications sur un fichier TAP décodé.

Groupe	Vérifications
Structure	Les cinq sections obligatoires sont présentes et non vides : `batchControlInfo`, `accountingInfo`, `networkInfo`, `auditControlInfo`, et `callEventDetails`.
Contrôle de lot	Le `sender` et le `recipient` ont 5 caractères. Le `fileSequenceNumber` a 5 caractères. Le `specificationVersionNumber` est `3`.
Comptabilité	Le `localCurrency` et le `tapCurrency` sont des codes ISO 4217 à 3 lettres. Le `tapDecimalPlaces` est un entier non négatif.
Arithmétique d'audit	Le nombre d'enregistrements est égal au nombre d'événements d'appel. Le total des charges, le total des taxes et le total des remises sont égaux aux sommes des enregistrements.

Gravité des erreurs

OmniRoam rapporte chaque résultat avec une gravité. Le fichier est valide uniquement lorsqu'il n'y a pas de résultats fatals.

Gravité	Effet
Fatal	Le fichier entier est rejeté.
Sévère	L'enregistrement individuel est rejeté. Le reste du fichier est accepté.
Avertissement	L'enregistrement est facturable. Le résultat est informatif.

Catégories de codes d'erreur

TD.57 regroupe les codes d'erreur en plages numériques.

Plage	Catégorie
10 à 19	Erreur de format ou de syntaxe.
20 à 29	Valeur hors limites.
30 à 49	Structure de groupe invalide.
50 à 59	Erreur ASN.1 ou d'encodage générale.
100 à 199	Incohérence entre les champs.
200 à 249	Erreur spécifique, comme un doublon, une erreur IOT ou des données anciennes.
9000 à 9999	Une erreur bilatérale, propriétaire, convenue entre deux partenaires.

Limites de la validation

OmniRoam valide les règles qu'il peut vérifier à partir du contenu du fichier seul. Certaines règles de TD.57 nécessitent des données de référence externes, et OmniRoam ne les vérifie pas.

- Listes de codes TADIG.
- Tarifs des accords de tarification inter-opérateurs et de roaming.
- Historique des doublons à travers des fichiers antérieurs.
- Recherches complètes de devises ISO 4217 et de numéros E.164.

Utilisez un accusé de réception de partenaire ou un fichier RAP pour confirmer le résultat du règlement. Voir Traitement RAP.

- Aperçu