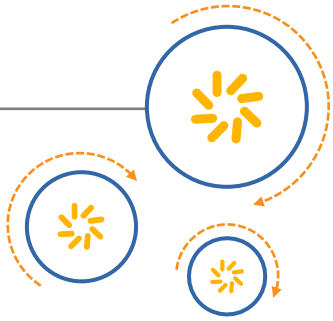
Qualcomm

Qualcomm Technologies, Inc.



DIRBS Core Release 16.0.0

User Guide

80-GD079-2 Rev. M

15/03/2021

Revision history

Revision	Date	Description
A	January 2018	Initial release
B	April 2018	Updates for DIRBS Core 7.0.0 release
C	May 2018	Updates for DIRBS Core 8.0.0 release
D	July 2018	Updates for DIRBS Core 8.0.1 release
E	September 2018	Updates for DIRBS Core 9.0.0 release
F	March 2019	Updates for DIRBS Core 10.0.0 release
G	October 2019	Updates for DIRBS Core 11.0.0 release
H	June 2020	Updates for DIRBS Core 12.0.0 release
I	September 2020	Updates for DIRBS Core 13.0.0 release
J	November 2020	Updates for DIRBS Core 14.0.0 release
K	January 2021	Updates for DIRBS Core 14.1.0 release
L	February 2021	Updates for DIRBS Core 15.0.0 release
M	March 2021	Updates for DIRBS Core 16.0.0 release

Contents

Figures

Tables

1 Introduction

The Device Identification, Registration & Blocking System (DIRBS) is a country-wide system deployed in cooperation between the country regulator, operators in that country, and a technology partner that supports deployment. The system checks, identifies, and discourages non-compliant devices by verifying the installed base of devices currently active in a market and continuing to monitor as new devices are activated.

DIRBS can verify that:

- Devices have properly allocated identifiers and type approval
- Devices are not duplicated or stolen
- Device importation takes place through legal channels

DIRBS consists of the DIRBS Core and a set of DIRBS Interface Systems that interface with DIRBS Core (see **Figure 1-1** and **Figure 1-2**).

E:DIRBS Interface Systems may be developed by third-party technology partner(s).

- All operators provide data to country's centralized DIRBS
- IMEIs are classified using configurable conditions
- Lists are generated for operators
- Reports are generated at operator and country levels
- Subsystems interface with core analysis engine

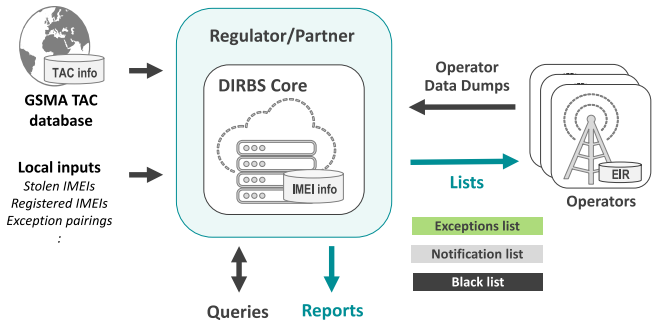


Figure 1-1 DIRBS Core

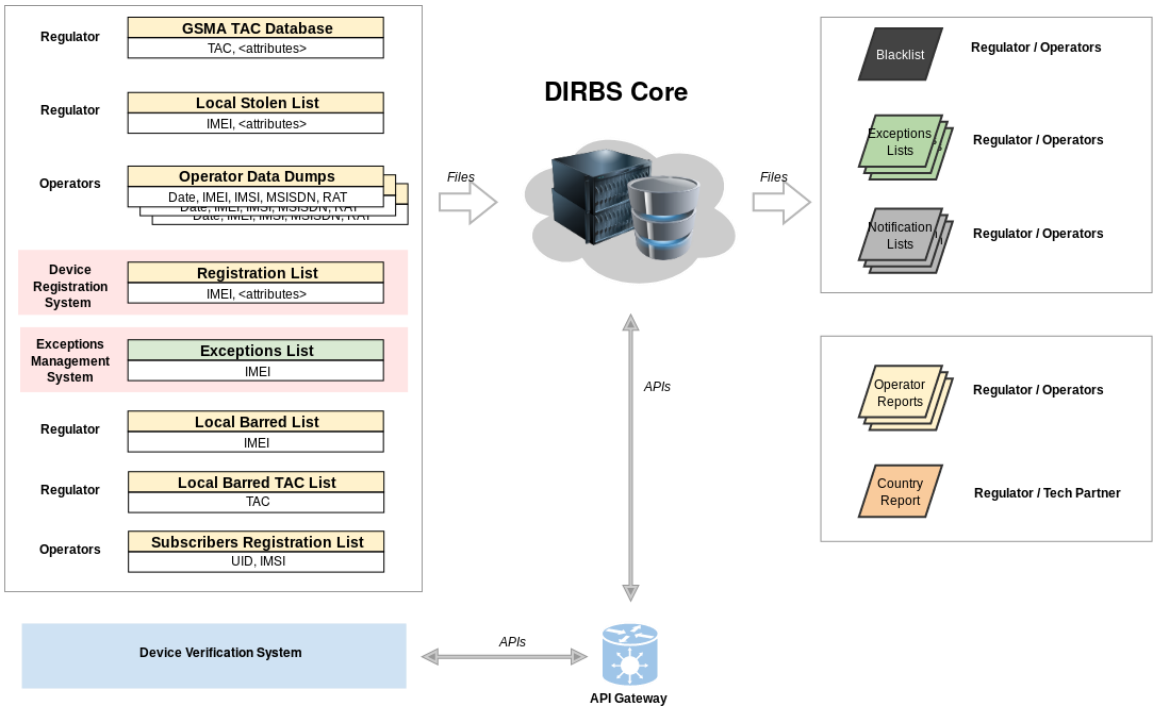


Figure 1-2 DIRBS input/output

2 Configuring DIRBS Core

A sample config file, `config.yml`, was distributed with this release. When DIRBS Core scripts run, they look for a file in `~/.dirbs.yml`. If that file is not found, look in the system location `/opt/dirbs/etc/config.yml`.

For the complete sample .yml file, see .

2.1 Copying the config file

To run a configuration using Docker, edit `./etc/config.yml`, and rebuild the Docker images.

If you are not using Docker:

1. Copy dirbs config to the appropriate location with:
   `cp /opt/dirbs/etc/config.yml ~/.dirbs.yml`

2.2 Database details

Table 2-1 Configuring database settings lists settings of the `postgresql` section in the sample config. These values are overridden by the environment variable if configured.

Table 2-1 Configuring database settings

Config file setting name	Environment variable name	Function
database	DIRBS_DB_DATABASE	Database name (an empty database on the first run)
host	DIRBS_DB_HOST	Host that the PostgreSQL server runs on
port	DIRBS_DB_PORT	PostgreSQL port if not running on standard port of 5432
user	DIRBS_DB_USER	Username (user should have all privileges specified on the database)
password	DIRBS_DB_PASSWORD	Password used to connect to the database

E:The database password can be configured in several ways and are dependent on the level of security required in your deployed system. The password may be configured in the following ways:

- In clear text in the `.dirbs.yml` file by configuring the password setting.
- The user's .pgpass file located in their home directory.
- The DIRBS_DB_PASSWORD environment variable is set. This overwrites values configured in the `.dirbs.yml` and the .pgpass file
- The password can be provided in the dirbs command-line option as `-db-password-prompt`.

2.3 Regional settings

DIRBS Core uses regional settings for reporting and input validation. Table 2-2 Configuring regional settings lists settings of the region section in the sample config.

Table 2-2 Configuring regional settings

Config file setting name	Value/ range	Function
name	String	Name used for the country level report
import_msisdn_data	True/False	Whether or not MSISDN data is present and should be imported for this region
import_rat_data	True/False	Whether or not RAT data is present and should be imported for this region
country_codes	Digits	▪ List of country codes for the region. ▪ Used to validate MSISDNs in operator data dumps
exempted_device_types	String	▪ Exempted_device_types contains a list of GSMA device types that do not require registration in this country. ▪ Specifiying a list of device types here will mean that the not_in_registration_list classification dimension will ignore IMEIs whose TACs correspond to the listed device types. They will also be ignored in the IMEI API's real-time registration check.
operators	See Section 2.3.1	See Section 2.3.1

2.3.1 Operator settings

The region config has an `operators` section. Table 2-3 Operator settings lists the settings for each operator section in the sample config.

Table 2-3 Operator settings

Setting	Value, range	Function
id	String	Operator ID (short string used to uniquely id operator)

name	String	Longer name for the operator, used in reports
mcc_mnc_pairs	Digits	MCC/MNC pairs for the operator ▪ Validate IMSIs during operator data import ▪ Determine which operator notifications about an offending subscriber should be sent to ▪ Determine which operators excepted IMEI-IMSI pairings should be sent to

2.4 Data purging

The sample config has a `data_retention` section that defines the number of months operator data can remain before it is deleted from the system and a `blacklist_retention` section that defines the number of days an IMEI is to remain in blacklist (see Table 2-4 Data retention settings).

E:The DIRBS Core system does not delete CSV data off disk. Purging only applies to data in the database. It is recommended that an operator of the DIRBS Core system write a small job to remove old CSV data if this is also required by regulation.

Table 2-4 Data retention settings

Setting	Value, range	Function
months_retention	Integer	▪ The number of months from the start of the current month that DIRBS Core retains data about a triplet seen in its DB. ▪ After this time, the triplet will be erased from the seen_triplet table. The IMEI continues to be stored after this date as it is needed for continued list generation, etc. All references to IMSI and MSISDN will be pruned after this date. ▪ If this value is set to two months and the current date is March 29, only the data in December or earlier will be purged from the database when the `dirbs-prune` command is run.
blacklist_retention	Integer	▪ The number of days for an IMEI to be in blacklisted. ▪ After this time, the IMEI will be expired from blacklist.

2.5 List generation settings

The sample config has a `list_generation` section containing a list of configurable settings related to list generation (see Table 2-5 List generation settings).

Table 2-5 List generation settings

Setting	Value, range	Function
lookback_days	Integer	The number of days that DIRBS Core looks back through data from the current date to determine

		IMSIs/MSISDNs which were associated with the notifiable IMEIs.
restrict_exceptions_list_to_blacklisted_imeis	True/False	▪ If true, the exception list contains only those IMEI-IMSI pairs where the IMEI is on the blacklist. ▪ By default, all IMEI-IMSI pairs of the pairing list are output to the exception list.
generate_check_digit	True/False	▪ If true, generates a check digit for IMEIs during list generation. ▪ Check digit will only be added to "valid IMEIs".
output_invalid_imeis	True/False	▪ If true, outputs only "valid" IMEIs. ▪ Valid IMEIs start with 14 digits (they will have 15 digits if the check digit append has been enabled).
notify_imsi_change	True/False	Weather to include those IMEIs for which IMSI is changed in notifications lists or not
include_barred_imeis_in_exceptions_lists	True/False	Weather to include IMEI(s) in exceptions lists who is in barred list or IMEI itself is in barred list. The default value is False

2.6 Report generation settings

The sample config has a `report_generation` section containing a list of configurable settings related to report generation process (see Table 2-6 Report generation settings).

Note: DIRBS Core Reports will be deprecated as the reporting will now be supported in DIRBS View.

Table 2-6 Report generation settings

Setting	Value, range	Function
blacklist_violations_grace_period_days	Integer	Used by blacklist violations and stolen list violations reports to give the MNO some processing time (in days) before an IMEI appearing on the network is considered a violation.

2.7 Multiprocessing settings

The sample config has a `multiprocessing` section that outlines how DIRBS Core uses multiple cores and database connections to speed up processing (see Table 2-7 Multiprocessing settings).

Table 2-7 Multiprocessing settings

Setting	Value, range	Function
max_local_cpus	Integer	▪ Maximum number of local processing blade workers to achieve DIRBS Core tasks ▪ Useful for pre-validation of large operator import jobs where we can run multiple instances of the pre-validator in parallel on different parts of the file ▪ Default is to use half of the available CPUs in the system
max_db_connections	Integer	▪ Maximum number of database connections to parallelize DIRBS Core tasks ▫ PostgresSQL 9.6 has support for parallelizing tasks internally and this setting does not affect parallelization for a single connection ▪ When PostgresSQL cannot parallelize a single query by itself, we use this number of workers to issue multiple queries at once on different connections ▫ Generally, scales very well, safe to set this high ▫ Should probably be set to roughly the number of disks in your RAID array
max_db_writers	Integer	▪ Number of I/O write-intensive DB connections to use at once ▫ Should be lower than max_db_connections ▫ For RAID-10, this should be set to about half of max_db_connections to indicate the paired nature of that type of RAID array

2.8 Operator data import validation thresholds

The sample config has an `operator_threshold` section containing a list of validation thresholds for operator data. These thresholds are used when importing operator data using the `dirbs-import` command. Table 2-8 Import validation thresholds lists the settings for these thresholds and their functions.

Table 2-8 Import validation thresholds

Setting	Value, range	Function
null_imei_threshold null_imsi_threshold null_msisdn_threshold null_rat_threshold	Decimal, 0.0-1.0	▪ Proportion of entries in the data allowed to have one or more of IMSI, MSISDN, IMEI, or RAT as null ▪ null_msisdn_threshould is ignored if import_msisdn_data is set to False ▪ null_rat_threshold is ignored if import_rat_data is set to False
null_threshold	Decimal, 0.0-1.0	▪ Proportion of the entries in the data allowed to have any column equal to NULL.

		▪ Only includes columns enabled in the import (MSISDN and RAT may be excluded)
unclean_imei_threshold	Decimal, 0.0-1.0	Proportion of non-NULL IMEIs in the data allowed to not start with 14 digits
unclean_imsi_threshold	Decimal, 0.0-1.0	Proportion of non-NULL IMSIs in the data allowed to not be 14-15 digits
unclean_threshold	Decimal, 0.0-1.0	Proportion of entries in the data allowed to have either an unclean IMEI or an unclean IMSI
out_of_region_imsi_threshold	Decimal, 0.0-1.0	Proportion of non-NULL IMSIs in the data allowed to have an MCC that does not match the configured region
out_of_region_msisdn_threshold	Decimal, 0.0-1.0	▪ Proportion of non-NULL MSISDNs in the data allowed to have a CC that does not match the configured region ▪ Ignored if MSISDN disabled
out_of_region_threshold	Decimal, 0.0-1.0	▪ Combined proportion of entries in the data allowed to have either a CC (IMSI) or MCC (MSISDN) that does not match the configured region ▪ Ignored if MSISDN disabled (same as the out-of-region IMSI check)
non_home_network_threshold	Decimal, 0.0-1.0	Proportion of entries in the data allowed to have an IMSI not starting with one of the MCC-MNC prefixes associated with the operator the data is being imported for
historic_imei_threshold	Decimal, 0.0-1.0	Minimum valid ratio of average daily IMEI count against historical daily IMEI count for a data dump to be considered valid
historic_imsi_threshold:	Decimal, 0.0-1.0	Minimum valid ratio of average daily IMSI count against historical daily IMSI count for a data dump to be considered valid
historic_msisdn_threshold	Decimal, 0.0-1.0	▪ Minimum valid ratio of average daily MSISDN count against historical daily MSISDN count for a data dump to be considered valid ▪ Ignored if MSISDN disabled
leading_zero_suspect_limit	Decimal 0.0-1.0	Limit for suspected leading zeros in operator data dump

2.9 Historic thresholds

The sample config has a `historic_thresholds` section that can validate new import row count against previously imported data for the same importer. Table 2-9 Historic threshold lists the settings for these thresholds and their functions.

Table 2-9 Historic threshold

Setting	Value, range	Function
gsma_threshold: import_size_variation_absolute:	 Integer Decimal, 0.0-1.0	▪ import_size_variation_absolute: The most an import can decrease in absolute row count before it is rejected

import_size_variation_percent: pairing_list_threshold: import_size_variation_absolute: import_size_variation_percent: stolen_list_threshold: import_size_variation_absolute: import_size_variation_percent: registration_list_threshold: import_size_variation_absolute: import_size_variation_percent: golden_list_threshold: import_size_variation_absolute: import_size_variation_percent: barred_list_threshold: import_size_variation_absolute: import_size_variation_percent: barred_tac_list_threshold: import_size_variation_absolute: import_size_variation_percent: subscribers_list_threshold: import_size_variation_absolute: import_size_variation_percent: association_list_threshold: import_size_variation_absolute: import_size_variation_percent:	Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0 Integer Decimal, 0.0-1.0	as invalid. By setting this variable to -1, this check will be disabled. ▪ import_size_variation_percent: The most an import can decrease in percentage row count before it is rejected as invalid. ▫ 0.75 indicates a new import must be at least 75% of the previous import's row count or it will be rejected. Therefore, setting this variable to 0 will disable this check.

2.10 Classification conditions

The sample config has a `conditions` section containing a list of conditions that classify IMEIs (see Table 2-10 Conditions settings).

Table 2-10 Conditions settings

Setting	Function
label	▪ A name/ID/key for the condition ▪ If label is changed, all previous classifications will be reset ▫ Likewise, if you change the dimensions but keep the condition label the same, existing classifications for that condition will be retained.
dimensions	▪ For dimension configuration details, see Section 3.5. ▪ A list of dimensions whose intersection forms the IMEI set result for the condition. Each dimension can take parameters specific to the dimension being used. Additionally, they all accept an invert property, which NOTs the result of the dimension by taking the all-time observed IMEIs list and subtracting the set of IMEIs returned by this dimension. ▪ Each dimension has: ▫ module: Python code module that implements condition. ▫ parameters: List of parameters supplied to dimension (available parameters depend on dimension). ▫ invert: Boolean stating whether results from this dimension should be inverted to produce a NOT result, i.e., inverting the GSMA Not Found dimension would return a list of all seen IMEIs that were found in the GSMA TAC DB.
grace_period_days	Number of days after DIRBS Core detects condition has been met by IMEI before it moves from the notification list to the blacklist.
blocking	▪ Boolean stating whether this condition contributes to list generation or is simply informational. ▪ Information conditions can be used to try out new modules or to tweak parameters.
reason	Human-readable reason string summarizing this condition (to be used in notification lists and blacklist).
max_allowed_matching_ratio	▪ The maximum ratio of IMEIs that can be matched before the condition fails because something is wrong. ▫ Example: Under normal circumstances, we might not expect more than 20% of all IMEIs to not be found in GSMA, so we can set this to 0.2. If more than this number of IMEIs match, we probably forgot to import the GSMA TAC DB, so we refuse to add the results of this condition to the classification_state table. Other conditions will still be added to the DB but the overall status of the dirbs-classify job will be non-zero (failure).

2.11 Logging settings

The sample config has a `logging` section containing a list of configurable settings related to logging output (see Table 2-11 Logging settings).

Table 2-11 Logging settings

Setting	Value, range	Function
level	String	▪ Minimum logging level to output log messages before ▪ Valid values are: ▫ debug ▫ info ▫ warn ▫ error
format	LogRecord Objects	Format string for log messages output by the application Example Format: format: '%(asctime)s - %(name)s - %(levelname)s - %(message)s'
show_statsd_messages	True/False	▪ Determines whether or not log messages related to StatsD should be output ▪ If enabled, all calls to StatsD will be logged for debugging
show_sql_messages	True/False	▪ Determines whether or not log queries made it to the SQL database ▪ If enabled (True), all queries will be logged for debugging (can result in very large output)
show_werkzeug_messages	True/False	▪ Determines whether or not log internal Werkzeug messages from the TAC/IMEI APIs ▪ Should almost always be set to False ▪ If enabled (True), all TAC/IMEI API queries will be logged (can resilt in large output)
log_directory	String	▪ Sets this directory to store the log files ▪ Directory must exist and be writable
file_prefix	String	Uncomment and set this value to prefix all log files created on this host with a prefix to distinguish them from other hosts
file_rotation_max_bytes	Integer	▪ Sets the number of bytes before a logfile is rotated ▪ If this or file_rotation_backup_count is 0, rotation is disabled
file_rotation_backup_count	Integer	Sets the number of old logs to keep

2.12 StatsD settings

The sample config has a `statsd` section containing a list of configurable settings related to forwarding application-defined metrics to a StatsD server for aggregation (see Table 2-12 StatsD Settings).

Table 2-12 StatsD Settings

Setting	Value, range	Function
hostname	String	▪ StatsD server hostname ▪ Overridden by the environment variable DIRBS_STATSD_HOST
port	Integer	▪ UDP port that StatsD server listens on for metrics ▪ Overridden by environment variable DIRBS_STATSD_PORT
prefix	True/False	▪ Prefix for all metrics collected from this instance ▫ Useful if there are multiple hosts or environments sending data to the same StatsD server and you want to differentiate them ▪ Overridden by the environment variable DIRBS_ENV

2.13 Data catalog settings

The sample config has a `catalog` section containing a list of configurable settings related to the data cataloging process (see Table 2-13 Data catalog settings).

Table 2-13 Data catalog settings

Setting	Value, range	Function
file_type	String	▪ Type of files contained within the specified paths ▪ Should match the keyword specified during `dirbs-import`, e.g. operator, gsma_tac, etc.
paths	Integer	▪ Directories and/or files to be harvested ▫ Sub-directories within the listed path are not traversed automatically ▫ Should be listed separately if files within them must be cataloged. ▪ Multiple paths can be defined for each file type and the path used should be absolute and globally unique
schema_filename:	String	▪ Schema file for data pre-validation (if enabled) ▪ Multiple prospectors can be defined for the same file_type if files exist across multiple schema versions
perform_prevalidation	String, True/False	▪ Set to true if pre-validation should be performed on the data files ▪ Enabling this can slow down the process if there are a lot of uncataloged files

2.14 Operational Settings

The sample config has an `operational` section containing a few configurable settings related to operation mode of DIRBS whitelist/restrictive mode or blacklist/permissive mode (see Table 2-14).

Table 2-14 Operational settings

Setting	Value, range	Function
activate_whitelist	Boolean	▪ To activate whitelist/restrictive mode, results in whitelist migrations and operations ▪ The default value is FALSE
restrict_whitelist	Boolean	▪ Weather to restrict whitelist sharing (with MNOs) or not

2.15 Broker Settings

The sample config has a `broker` section containing a list of configurable settings related to KAFKA and operators (see Table 2-15).

Table 2-15 Broker settings

Setting	Value, range	Function
kafka	section	▪ To specify KAFKA broker settings for whitelist 1. **hostname:** Host name or IP of the broker 2. **port:** Port of the broker host 3. **topic:** Topic name to subscribe to. It is used for DIRBS to receive notification and requests from the configured operators/MNOs 4. **security_protocol:** To use security protocol mode, currently two modes are available, PLAINTEXT and SSL. The default value is set to PLAINTEXT. 5. **client_certificate:** Path to the client certificate signed by the CA, which will be used for the SSL connection. The file should be in ".pem" format. 6. **client_key:** Path to the client key file. The file should be in ".pem" format. 7. **caroot_certificate:** Path to the CARoot certificate file. The file should be in ".pem" format. 8. **skip_tls_verifications:** Weather to skip TLS verification or not, default value is FALSE. It is recommended as FALSE in production environment.
operators	section	▪ To configure operators for DIRBS whitelist system. Multiple operators can be configured under this section. Following are the acceptable configuration parameters: 1. **id:** ID of a particular operator/MNO 2. **name:** Human readable name of the operator/MNO 3. **topic:** Name of the KAFKA topic used by the particular operator

2.16 Redis Settings

Table 2-16 Redis settings

Setting	Value, range	Function
hostname	String	The hostname for the Redis server. Overriding by environment variable DIRBS_REDIS_HOST if set. The default hostname is set to localhost if not set.
port	Integer	The port number that the Redis server is listening on. Overridden by environment variable DIRBS_REDIS_PORT if set. The default port is 6379 if not set.
password	String	The password for the redis server. Overriding by environment variable DIRBS_REDIS_PASSWORD if set. The default value is none.
db	String	The redis db instance to be used as cache. Overridden by DIRBS_REDIS_DB if set. The default is set to instance 0.
cache_timeout	Integer	Cache timeout in seconds, Overridden by environment variable DIRBS_REDIS_CACHE_TIMEOUT if set. The default value is 5 minutes.

3 Operating DIRBS Core

E:If you are not using Docker, the following commands only work when the DIRBS Core virtual environment is activated. This must be done each time you log into the machine or start a new shell.

To activate the virtual environment, run:

```
source <install_path>/bin/activate
```

3.1 Data in the Docker instance

If you run the Data Processing blade image, the `/data` directory is marked as a persistent data volume in Docker. Data in this directory will persist after a container has been destroyed.

For more information on data volumes, see https://docs.docker.com/engine/tutorials/dockervolumes/.

When the Data Processing blade image is created, the entry point script (entrypoint.sh) populates the `/data` directory with the required folder structure, along with the correct permissions. This script reads a list of operators from the DIRBS_OPERATOR environment variable and creates folders for each if they are missing.

There are a couple options to get data into the container:

- Bind-mount `/data` to a directory on the host machine using the `-v` option to `docker run`. If using this approach, ensure that the directory has the correct permissions and can be written to by the docker user. The entry point creates the requisite folders in this directory so it can be empty to begin with.
- Use commands `scp` or `ssh` to copy data into the container. Once it is in there, it will persist across container builds due to the persistent nature of the data volume.

3.2 Managing database schema – `dirbs-db`

`dirbs-db` manages the database schema version deployed to the PostgreSQL server.

For information on the `dirbs-db` command and its available subcommands, run:

```
dirbs-db --help
```

```
Usage: dirbs-db [OPTIONS] COMMAND [ARGS]...

  DIRBS script to intiliaze, configure and upgrade the PostgreSQL schema.

Options:
  --version               Show the version and exit.
  -v, --verbose           Print debug console output - file output is
                          unaffected.
  --db-password-prompt    If set, will prompt the user for a PostgreSQL
```



```
                          password rather than reading from config.
  --db-user TEXT          The PostgreSQL DB database user to connect as.
  --db-name TEXT          The PostgreSQL DB database name to connect to.
  --db-port INTEGER       The PostgreSQL DB port to connect to.
  --db-host TEXT          The PostgreSQL DB host to connect to.
  --statsd-prefix TEXT    The environment prefix to prepend to all StatsD
                          metrics.
  --statsd-port INTEGER   The StatsD port to connect to on the configured host.
  --statsd-host TEXT      The StatsD host to send metrics to.
  --help                  Show this message and exit.

Commands:
  check                     Checks whether DB schema matches software DB...
  repartition               Repartition DIRBS Core table into a new number of...
  install                   Installs latest schema on clean DB instance.
  install_roles             Creates DIRBS Core PostgreSQL base roles if...
  upgrade                   Upgrades the current DB schema to the version...
```

3.2.1 Creating DIRBS Core PostgreSQL roles – `dirbs-db install_roles`

E:Roles must be created and installed prior to running the `dirbs-db install` and `dirbs-db upgrade` commands. For detailed instructions on installing and configuring a new database, see *DIRBS Core Release 16.0.0 Installation Guide*.

```
dirbs-db --db-user <username> --db-password-prompt install_roles
```

where

- `<username>` is the name of the user with the CREATEROLE privilege

3.2.2 Installing schema into a clean database – `dirbs-db install`

To install the DIRBS Core schema into a clean database, run:

```
dirbs-db --db-user <username_of_power_user> --db-password-prompt install
```

This command only works on a clean database. You can force-install the schema into a non-clean database using `-force` flag, but this is dangerous as it may leave the database in an inconsistent state where future migration scripts fail.

Note: If `activate_whitelist` parameter is set to TRUE under the operational configuration section, the `dirbs-db install` command will install schema and tables for the DIRBS whitelist processing blades.

For help on the options available to `dirbs-db install`, run:

```
dirbs-db install --help
```

```
Usage: dirbs-db install [OPTIONS]

  Installs latest schema on clean DB instance.

Options:
  --help  Show this message and exit.
```

3.2.3 Checking status of the database schema – `dirbs-db check`

`dirbs-db check` displays which schema version is currently deployed to the PostgreSQL database, and which version is required by the installed code.

To run this check, run:

```
dirbs-db check
```

3.2.4 Upgrading the database schema – `dirbs-db upgrade`

It is recommended to take a database backup before attempting these steps in case something goes wrong. Downgrades are not possible.

`dirbs-db upgrade` upgrades the currently installed database schema in PostgreSQL to the version required by the installed software.

To run the upgrade, run:

```
dirbs-db --db-user <power_user> --db-password-prompt upgrade
```

where

- `<power_user>` is a user that has been `GRANT`'ed the `dirbs_core_poweruser` role

The upgrade script determines which version of the schema is required and automatically runs SQL migration scripts to upgrade the schema.

3.3 Importing data into DIRBS Core – `dirbs import`

This section describes how to import data into the DIRBS Core using `dirbs-import` functionality.

While the DIRBS System does not enforce a specific data import order, Figure 3-1 describes the recommended steps involved in importing data.

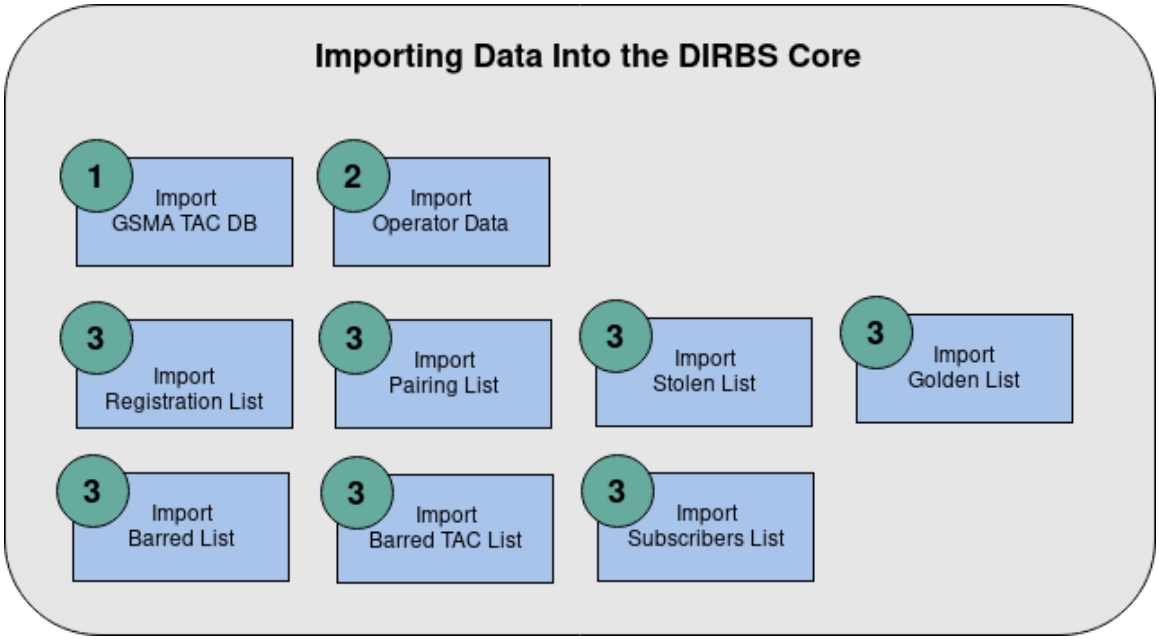


Figure 3-1 Importing data into DIRBS

During normal system operation, it is expected that jobs are scheduled and configured to periodically import these data files.

Depending on the analysis that you are performing you may or may not require the import of the files listed in the diagram above.

For information on the `dirbs-import` and its available subcommands, run:

```
dirbs-import --help
```

```
Usage: dirbs-import [OPTIONS] COMMAND [ARGS]...

  DIRBS script to import data into DIRBS Core PostgreSQL database.

Options:
  --version                       Show the version and exit.
  -v, --verbose                   Print debug console output - file output is
                                  unaffected.
  --db-password-prompt            If set, will prompt the user for a
                                  PostgreSQL password rather than reading from
                                  config.
  --db-user TEXT                  The PostgreSQL DB database user to connect
                                  as.
  --db-name TEXT                  The PostgreSQL DB database name to connect
                                  to.
  --db-port INTEGER               The PostgreSQL DB port to connect to.
  --db-host TEXT                  The PostgreSQL DB host to connect to.
  --statsd-prefix TEXT            The environment prefix to prepend to all
                                  StatsD metrics.
  --statsd-port INTEGER           The StatsD port to connect to on the
                                  configured host.
```



```
  --statsd-host TEXT              The StatsD host to send metrics to.
  --max-db-writers INTEGER        The maximum write-intensive DB connections
                                  to use concurrently during this job.
  --max-db-connections INTEGER    The maximum DB connections to use
                                  concurrently during this job.
  --max-local-cpus INTEGER        The maximum number of local CPUs to use
                                  concurrently during this job.
  --batch-size INTEGER            Size of batches to import into DB, in lines.
  --no-cleanup                    If set, intermediate split data files and
                                  the staging table will not be deleted so
                                  that they can inspected.
  --extract-dir DIRECTORY         Directory to extract contents of .zip file
                                  into (same directory as input file by
                                  default).
  --prevalidator-path PATH        The path to the CSV pre-validator
                                  executable.
  --prevalidator-schema-path DIRECTORY
                                  The path to the directory where the CSV pre-
                                  validator schema are stored.
  --help                          Show this message and exit.

Commands:
  golden_list         Import the Golden list data found in INPUT...
  gsma_tac            Import the GSMA TAC DB data found in INPUT...
  operator            Import the CSV operator data found in INPUT...
  pairing_list        Import the Pairing List data found in INPUT...
  registration_list   Import the Registration list data found in...
  stolen_list         Import the Stolen List data found in INPUT...
  barred_list         Import the Barred List data found in INPUT...
  barred_tac_list     Import the Barred TAC List data found in...
  subscribers_
  registration_list   Import the Subscribers Registration List…
  association_list    Import the Association List...
```

3.3.1 GSMA TAC DB – `dirbs-import gsma_tac`

To import a .zip version of the GSMA TAC database, run:

```
dirbs-import gsma_tac <gsma_zip_file>
```

The .zip file is expected to contain a .txt file where the columns are pipe-separated. It is run through the CSV pre-validator `GoldenListSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

Table 3-1 gsma_tac fields and format lists the expected header columns and format. These fields can be in any order and case.

Table 3-1 gsma_tac fields and format

Field	Mandatory (M) Optional (O)	Expected format
tac	M[1]	Integer, Length 8
manufacturer	O	String, Length (1-128) [2]
model name	O	String, Length (1-1024) [2]
bands	O	String, Length (1-4096) [2]
allocation date	O	▪ Day-Month-Year format[2], e.g., 26-Apr-2016

		▫ Day: 0 – 31 (must correspond to days in month/year) ▫ Month: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec ▫ Year: 19XX, 20XX
optional fields	O	String, Length (1-4096) [2]
1 The tac field must not be empty. 2 Empty string and whitespace allowed.		

The GSMA TAC DB importer can accept additional columns, which will be imported into an `optional_fields` column in the DB and returned by the TAC API without changes. No validation or processing takes place on these optional fields.

The GSMA TAC DB also performs a historic validation check as specified in section 2.9. If this check fails, the import is rejected. This check can be disabled with `--disable-historic-check`.

For help on all the options available to `dirbs-import gsma_tac`, run:

```
dirbs-import gsma_tac --help
```

```
Usage: dirbs-import gsma_tac [OPTIONS] INPUT_FILE

  Import the GSMA TAC DB data found in INPUT into the PostgreSQL database.

Options:
  --disable-historic-check    Skip checking the size of this import against
                              the currently stored data.
  --disable-duplicates-check  Skip checking for duplicate rows in this file
                              and failing if there any.
  --help                      Show this message and exit.
```

3.3.2 operator data dumps – `dirbs-import operator`

To import a .zip version of an operator data dump, run:

```
dirbs-import operator <operator_id> <operator_zip_file>
```

The .zip file is expected to contain a ‘,’ comma separated .csv file containing the operator data. It is run through the CSV pre-validator `OperatorImportSchema_v2.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

Table 3-2 operator data fields and format lists the expected header columns and format. These fields can be in any order and case.

Table 3-2 operator data fields and format

Field	Mandatory (M) Optional (O)	Expected format
date	M	▪ yyyymmdd format starting with year 2000, e.g., 20160703 ▫ yy: 20XX, where X=0-9 ▫ mm: 00-12 ▫ dd: 00 – 31 (must correspond to days in month/year)
imei	O	▪ Digits, Length (1 – 16) [1]

		▫ Valid digits: 0-9, A-F, a-f, *, # ▫ 14 leading digits for good IMEI Records (check digit/software version is stripped on import)
imsi	O	▪ Digits, Length (1-15) [1] ▫ Valid digits: 0-9
msisdn	O	▪ Digits, Length (1-15) [1] ▫ E.164 format
rat	O	▪ Digits, Length (3) [1] ▫ Within range of 001-007 or 101-105
1 Empty string and whitespace allowed but will be stored as NULL.		

`operator_id` is a string or number that uniquely identifies the operator which must match one of the operator IDs in the config .yml file. Data import will fail and will not be imported if the Operator ID does not match.

E:If operator IDs are modified/replaced after successfully importing operator data, that data will still be included in the country-level reporting and in blacklist generation. However, notification lists will not be generated for the previously replaced operatorID.

The default behavior of the operator data importer expects data to contain RAT information. If a data dump does not contain RAT information, it can be imported using the `--disable-rat-import`.

Validation checks

Operator data performs multiple validation checks during import. Table 3-3 Validation checks lists the validation checks and their functions.

Table 3-3 Validation checks

Check	Function
CSV pre-validation	Ensures input CSV conforms to the operator data schema
Filename checks	Ensures .zip file conforms to the required filename format (`<operator_id>_<startdate>_<enddate>.zip`) and that the .csv filename within the .zip also conforms to that filename format
Date checks	Ensures `connection_date` field in the CSV data falls within the date range specified by the filename
Null checks	Ensures CSV data does not contain too many rows with blanks for IMEIs, IMSIs, MSISDNs, and/or RAT values
Clean checks	Ensures CSV data does not contain too many rows with invalid characters in the IMEI, IMSI or MSISDN
Leading zero checks	Ensures leading zeroes have not been stripped from the IMEIs in the CSV data
Region checks	Ensures that not too many rows contain IMSIs or MSISDNs with out-of-region CC and MCC values
Home network checks	Ensures that not too many rows contain IMSIs with an MCC-MNC prefix not associated with the operator
Historic checks	▪ Ensures data is consistent with previous imports from the same operator ▪ Performed based on previously generated reports

	▫ If reporting was never performed, historic checks will not be performed ▫ These checks compare the average daily counts for IMEI, IMSIs, and MSISDNs against historical counts

Most of these checks can be disabled with command-line options. For help on all the options available to `dirbs-import operator`, run:

```
dirbs-import operator --help
```

```
Usage: dirbs-import operator [OPTIONS] OPERATOR_ID INPUT_FILE

  Import the CSV operator data found in INPUT into the PostgreSQL database.

  OPERATOR_ID is an ID up to 16 characters to unique identify the operator.

Options:
  --disable-leading-zero-check  Skip checking if the import data appears to
                                have lost leading zeros.
  --disable-null-check          Skip checking the ratio of IMSIs, MSISDNs,
                                IMEIs and RATs that are NULL.
  --disable-clean-check         Skip checking the ratio of IMEIs and IMSIs
                                that are the wrong length or contain invalid
                                characters.
  --disable-region-check        Skip checking the ratio of MSISDNs and IMSIs
                                that have out of region cc and mcc values.
  --disable-home-check          Skip checking the ratio of and IMSIs that have
                                out of region mcc and mnc pair values.
  --disable-msisdn-import       Skip importing MSISDN field even if it does
                                exist in input data.
  --disable-rat-import          Skip importing RAT field if it does not exist
                                in input data.
  --disable-historic-check      Skip checking the size of this import against
                                the currently stored data.
  --help                        Show this message and exit.
```

3.3.3 golden list data – `dirbs-import golden_list`

To import a .zip version of the complete golden list, run:

```
dirbs-import golden_list <golden_list_zip_file>
```

The golden list identifies IMEIs of high-ranking officials to be excluded from being blocked.

TION:Any IMEIs added to the golden list will never be blocked.

The .zip file is expected to contain a .csv file containing the list of golden list data. It is run through the CSV pre-validator `GoldenListSchemaPreHashedData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The golden list can be imported with MD5 pre-hashed IMEIs or non-hashed IMEI. When hashing the IMEIs, it is expected to hash a 14-digit IMEI (see Section 7.6)

Table 3-4 golden list MD5 pre-hashed fields and format and Table 3-5 golden list fields and format list the expected header columns and format of a golden list.

Table 3-4 golden list MD5 pre-hashed fields and format

Field	Mandatory (M) Optional (O)	Expected format
golden_imei	M	Hex String, Length (32), MD5 Encrypted Valid characters: 0-9, A-F, a-f, *, #

Table 3-5 golden list fields and format

Field	Mandatory(M) Optional(O)	Expected format
golden_imei	M	Digits, Length (1 – 16) Valid digits: 0-9, A-F, a-f, *, # Do not remove leading zeros

For help on all the options available to `dirbs-import golden list`, run:

```
dirbs-import golden_list --help
```

```
Usage: dirbs-import golden_list [OPTIONS] INPUT_FILE

  Import the Golden list data found in INPUT into the PostgreSQL database.

  NOTE: Use caution when adding entries to the Golden list, as any IMEIs
  added to this list will never be blocked.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --pre-hashed TEXT              DANGEROUS: The input file contains normalized
                                 IMEIs that have already been hashed using the
                                 MD5 algorithm. If IMEIs have not been
                                 normalized or hashed according to DIRBS Core
                                 rules, the IMEIs in the imported list may not
                                 be correctly excluded from being blocked.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.3.1 Golden list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include ‘add’, ‘remove’.

The golden list delta import functionality can be invoked by the command line option:

```
dirbs-import golden_list <golden_list_delta_zip_file>
```

The .zip file is expected to contain a .csv file containing the delta golden list data. It is run through the CSV pre-validator `GoldenListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The `golden_imei` field is used as a key column to uniquely identify an entry in the list (see Section 7.6).

Sample delta .csv file

```
golden_imei,change_type
622222222222222,add
633333333333333,remove
```

3.3.4 Local stolen list data – `dirbs-import stolen_list`

To import a .zip version of the local stolen list, run:

```
dirbs-import stolen_list <stolen_list_zip_file>
```

The stolen list inputs IMEIs of stolen devices and the reported stolen date.

The .zip file is expected to contain a .csv file containing the list of data. It is run through the CSV pre-validator `StolenListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-6 stolen list fields and format lists the expected header columns and format. These fields can be in any order and case.

Table 3-6 stolen list fields and format

Field	Mandatory (M) Optional (O)	Expected format
imei	M	▪ Digits, Length (1 – 16) ▫ Valid digits: 0-9,A-F,a-f,*,# ▫ 14 leading digits for good IMEI Records (check digit/software version is stripped on import)
reporting_date	M	▪ yyyymmdd format[1], e.g., 20160703 ▫ Starting with year 2000 ▫ yy: 20XX, where X=0-9 ▫ mm: 00-12 ▫ dd: 00 – 31 (must correspond to days in month/year)
status	O	▪ Accepts any string but blacklist means that the IMEI is treated as the process is complete and the device should be blocked.

For help on all the options available to `dirbs-import stolen_list`, run:

```
dirbs-import stolen_list –help
```

```
Usage: dirbs-import stolen_list [OPTIONS] INPUT_FILE

  Import the Stolen List data found in INPUT into the PostgreSQL database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
```



```
                              updates in delta list are already in DB.
  --delta                     Switch to delta import mode.
  --help                      Show this message and exit.
```

3.3.4.1 Stolen list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include 'add', 'remove' or 'update'.

The stolen list delta import functionality can be invoked by the command line option:

`dirbs-import stolen_list <stolen_list_delta_zip_file>`

The .zip file is expected to contain a .csv file containing the delta stolen list data. It is run through the CSV pre-validator `StolenListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The `imei` field is used as a key column to uniquely identify an entry in the list (see Section 7.6).

Sample delta .csv file

```
imei,reporting_date,status,change_type
622222222222222,20180909,blacklist,add
633333333333333,20180909,pending,remove
644444444444444,20180909,abcd,update
```

3.3.5 Pairing list data – `dirbs-import pairing_list`

To import a .zip version of the complete pairing list, run:

`dirbs-import pairing_list <pairing_list_zip_file>`

The pairing list inputs IMEI-IMSI-MSISDN triplets that will be excluded from blocking.

The .zip file is expected to contain a .csv file containing the list of IMEI-IMSI-MSISDN triplets. It is run through the CSV pre-validator `PairingListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-7 pairing list fields and format lists the expected header columns and format. These fields can be in any order and case.

Table 3-7 pairing list fields and format

Field	Mandatory (M) Optional (O)	Expected format
imei	M	Digits, Length (1 – 16) Valid digits: 0-9, A-F, a-f, *, # 14 leading digits for good IMEI Records (check digit/software version is stripped on import)
imsi	M	Digits, Length (1-15) Valid digits: 0-9
msisdn	M	Digits, Length (1-15) Valid digits: 0-9

For help on all the options available to `dirbs-import pairing_list`, run:


```
dirbs-import pairing_list --help
```

```
Usage: dirbs-import pairing_list [OPTIONS] INPUT_FILE

  Import the Pairing List data found in INPUT into the PostgreSQL database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.5.1 Pairing list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include 'add' or 'remove'.

The pairing list delta import functionality can be invoked by the command line option:

```
dirbs-import pairing_list <pairing_list_delta_zip_file>
```

The .zip file is expected to contain a .csv file containing the delta pairing list data. It is run through the CSV pre-validator `PairingListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The `imei, imsi` fields are a key column to uniquely identify an entry in the list (see Section 7.6).

Sample delta .csv file

```
imei,imsi,msisdn,change_type
00333333333333,003333333333333,003333333344444,add
012222222222222,012222222222222,012222222233333,add
02333333333333,023333333333333,023333333366666,remove
```

3.3.6 Registration list data – `dirbs-import registration_list`

To import a .zip version of the complete import list, run:

```
dirbs-import registration_list <registration_list_zip_file>
```

The purpose of the pairing list is to input IMEIs that have been registered.

The .zip file is expected to contain a .csv file containing the list of approved IMEIs. It is run through the CSV pre-validator `RegistrationListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-8 registration list fields and format

Field	Mandatory (M) Optional (O)	Expected format
approved_imei	M	Digits, Length (1 – 16) Valid digits: 0-9,A-F,a-f,*,#
make	O	String, Length (1-4096) [2]
model	O	String, Length (1-4096) [2]
status	O	String, Length (1-4096) [2]
model_number	O	String, Length (1-4096) [2]
brand_name	O	String, Length (1-4096) [2]
device_type	O	String, Length (1-4096) [2]
radio_interface	O	String, Length (1-4096) [2]
device_id	M	Digits, Valid Digits: 0-9, A-F, a-f

For help on all the options available to `dirbs-import registration_list`, run:
`dirbs-import registration_list --help`

```
Usage: dirbs-import registration_list [OPTIONS] INPUT_FILE

  Import the Registration list data found in INPUT into the PostgreSQL
  database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.6.1 Registration list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include ‘add’ or ‘remove’.

The pairing list delta import functionality can be invoked by the command line option:
`dirbs-import registration_list <registration_list_delta_zip_file>`

The .zip file is expected to contain a .csv file containing the delta registration list data. It is run through the CSV pre-validator `RegistrationListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.
The `approved_imei` field is used as a key column to uniquely identify an entry in the list (see Section 6.5).

Sample delta .csv file

```
approved_imei, make, model, status, model_number, brand_name, device_type,
radio_interface, device_id, change_type
```



```
10000000000000,,,,,,,222,add
10000000000001,,,,,,,222,remove
10000000000002,,,,,,,123,add
```

3.3.7 Barred list data – `dirbs-import barred_list`

To import a .zip version of the complete import list, run:

```
dirbs-import barred_list <barred_list_zip_file>
```

The purpose of the barred list is to input IMEIs that have been denied access from the network irrespective of any reason. An authority or regulator can avail this feature to solve different business problems. These IMEI(s) are also classified using `exists_in_barred_list` dimension.

The .zip file is expected to contain a .csv file containing the list of barred IMEIs. It is run through the CSV pre-validator `BarredListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-9 barred list fields and format

Field	Mandatory (M) Optional (O)	Expected format
imei	M	Digits, Length (1 – 16) Valid digits: 0-9, A-F, a-f, *, #

For help on all the options available to `dirbs-import barred_list`, run:

```
dirbs-import barred_list --help
```

```
Usage: dirbs-import barred_list [OPTIONS] INPUT_FILE

  Import the Barred list data found in INPUT into the PostgreSQL
  database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.7.1 Barred list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include ‘add’ or ‘remove’.

The barred list delta import functionality can be invoked by the command line option:

```
dirbs-import barred_list <barred_list_delta_zip_file>
```


The .zip file is expected to contain a .csv file containing the delta barred list data. It is run through the CSV pre-validator `BarredListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The `imei` field is used as a key column to uniquely identify an entry in the list (see Section 6.5).

Sample delta .csv file

```
imei, change_type
10000000000000,add
10000000000001,remove
10000000000002,add
```

3.3.8 Barred TAC list data – `dirbs-import barred_tac_list`

To import a .zip version of the complete import list, run:

```
dirbs-import barred_tac_list <barred_tac_list_zip_file>
```

The purpose of the barred TAC list is to input TACs to which associated IMEIs have been denied access from the network irrespective of any reason. An authority or regulator can avail this feature to solve different business problems. IMEIs belong to these TACs are also classified using `is_barred_tac` dimension.

The .zip file is expected to contain a .csv file containing the list of barred TACs. It is run through the CSV pre-validator `BarredTacListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-10 barred tacs list fields and format

Field	Mandatory (M) Optional (O)	Expected format
TAC	M	Positive Integer, Length (8)

For help on all the options available to `dirbs-import barred_tac_list`, run:

```
dirbs-import barred_tac_list --help
```

```
Usage: dirbs-import barred_tac_list [OPTIONS] INPUT_FILE

  Import the Barred TAC list data found in INPUT into the PostgreSQL
  database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```


3.3.8.1 Barred TAC list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include 'add' or 'remove'.

The barred tac list delta import functionality can be invoked by the command line option:

```
dirbs-import barred_tac_list <barred_tac_list_delta_zip_file>
```

The .zip file is expected to contain a .csv file containing the delta tac list data. It is run through the CSV pre-validator `BarredListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The tac field is used as a key column to uniquely identify an entry in the list (see Section 6.5).

Sample delta .csv file

```
tac, change_type
10000012,add
10000022,remove
10000033,add
```

3.3.9 Subscribers list data – `dirbs-import subscribers_registration_list`

To import a .zip version of the complete import list, run:

```
dirbs-import subscribers_registration_list <subscribers_list_zip_file>
```

The purpose of the subscribers registration list is to input the pair of IMSI-UID into the DIRBS Core. This list is used to assist in duplicate detection.

The .zip file is expected to contain a .csv file containing the list of IMSIs and UIDs. It is run through the CSV pre-validator `SubscribersRegistrationListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-11 subscribers list fields and format

Field	Mandatory (M) Optional (O)	Expected format
uid	M	Digits, Length (1 – 20) Valid digits: 0-9, A-F, \, -
imsi	M	Digits, Length (1-15) Valid digits: 0-9

For help on all the options available to `dirbs-import subscribers_registration_list`, run:

```
dirbs-import subscribers_registration_list --help
```

```
Usage: dirbs-import subscribers_registration_list [OPTIONS] INPUT_FILE

  Import the Subscribers Registration list data found in INPUT into the PostgreSQL
  database.
```



```
Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.9.1 Subscribers list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include 'add' or 'remove'.

The subscribers registration list delta import functionality can be invoked by the command line option:

```
dirbs-import subscribers_registration_list
<subscribers_list_delta_zip_file>
```

The .zip file is expected to contain a .csv file containing the delta subscribers list data. It is run through the CSV pre-validator `SubscribersRegistrationListDeltaSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The combination of uid and imsi field is used as a key column to uniquely identify an entry in the list (see Section 6.5).

Sample delta .csv file

```
uid,imsi, change_type
abshh837388,10000000000000,add
63sjdk892ks,10000000000001,remove
8829sksjskd,10000000000002,add
```

3.3.10 Association list data – `dirbs-import device_association_list`

To import a .zip version of the complete import list, run:

```
dirbs-import device_association_list <association_list_zip_file>
```

The purpose of the device association list is to input the pair of IMEI-UID into the DIRBS Core. This list is used to assist in duplicate detection.

The .zip file is expected to contain a .csv file containing the list of IMEIs and UIDs. It is run through the CSV pre-validator `DeviceAssociationListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-12 association list fields and format

Field	Mandatory (M) Optional (O)	Expected format
uid	M	Digits, Length (1 – 20) Valid digits: 0-9,A-F,\,-
imei	M	Digits, Length (1 – 16) Valid digits: 0-9, A-F, a-f, *, #

For help on all the options available to `dirbs-import device_association_list`, run:

```
dirbs-import device_association_list --help
```

```
Usage: dirbs-import device_association_list [OPTIONS] INPUT_FILE

  Import the Device Association list data found in INPUT into the PostgreSQL
  database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.10.1 Association list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include ‘add’ or ‘remove’.

The subscribers registration list delta import functionality can be invoked by the command line option:

```
dirbs-import device_association_list <association_list_delta_zip_file>
```

The .zip file is expected to contain a .csv file containing the delta association list data. It is run through the CSV pre-validator `DeviceAssociationListDeltaSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The combination of uid and imsi field is used as a key column to uniquely identify an entry in the list (see Section 6.5).

Sample delta .csv file

```
uid,imei, change_type
abshh837388,10000000000000,add
63sjdk892ks,10000000000001,remove
8829sksjskd,10000000000002,add
```

3.3.11 Monitoring list data – `dirbs-import monitoring_list`

To import a .zip version of the complete import list, run:


```
dirbs-import monitoring_list <monitoring_list_zip_file>
```

The purpose of the monitoring list is to input IMEIs that needs to be monitored across the network for any reason. An authority or regulator can avail this feature to solve different business problems. These IMEI(s) are also classified using `exists_in_monitoring_list` dimension.

The .zip file is expected to contain a .csv file containing the list of IMEIs. It is run through the CSV pre-validator `MonitoringListSchema.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core (see Section 7.6).

Table 3-13 monitoring list fields and format

Field	Mandatory (M) Optional (O)	Expected format
imei	M	Digits, Length (1 – 16) Valid digits: 0-9,A-F,a-f,*,#

For help on all the options available to `dirbs-import monitoring_list`, run:

```
dirbs-import monitoring_list --help
```

```
Usage: dirbs-import monitoring_list [OPTIONS] INPUT_FILE

  Import the Monitoring list data found in INPUT into the PostgreSQL
  database.

Options:
  --disable-historic-check       Skip checking the size of this import against
                                 the currently stored data.
  --disable-delta-adds-check     If in delta mode, disable verification that
                                 adds in delta list are not already in DB.
  --disable-delta-removes-check  If in delta mode, disable verification that
                                 removes in delta list are already in DB.
  --disable-delta-updates-check  If in delta mode, disable verification that
                                 updates in delta list are already in DB.
  --delta                        Switch to delta import mode.
  --help                         Show this message and exit.
```

3.3.11.1 Monitoring list delta import

The delta list import feature was designed to allow regulators to supply changes to lists in a file rather than the complete file every time. These changes include ‘add’ or ‘remove’.

The monitoring list delta import functionality can be invoked by the command line option:

```
dirbs-import monitoring_list <monitoring_list_delta_zip_file>
```

The .zip file is expected to contain a .csv file containing the delta monitoring list data. It is run through the CSV pre-validator `MonitoringListDeltaSchemaData.csvs` located at `/opt/dirbs/etc/schema` to ensure it conforms to the expected format by DIRBS Core.

The `imei` field is used as a key column to uniquely identify an entry in the list (see Section 6.5).

Sample delta .csv file

```
imei, change_type
10000000000000,add
10000000000001,remove
10000000000002,add
```


3.4 Automating import of new files

3.4.1 Sample Makefiles and Jenkins

The `dirbs-import` command works very well for a single file, but DIRBS Core also provides the ability to monitor a directory for new files and automatically import them. This functionality is provided by the standard UNIX utility `make`.

We have provided Makefiles in the release distributables under "etc/makefiles". Once installed, these Makefiles are located at `/opt/dirbs/etc/makefiles`. There is a different Makefile for each type of import.

Whenever `make` is invoked, it looks for files where there is no corresponding `.processed` file, or where the source file is newer than the `.processed` file. For each file it finds that matches the previous criteria, `dirbs-import` imports the file and creates a `.processed` file.

This approach is very flexible and can be integrated with crontabs or with a more sophisticated approach using Jenkins. In both approaches, either a crontab entry or Jenkins job would be created for each type of import and for each operator.

Sample Makefile invocations

- To import GSMA data:
  ```
  make -f /opt/dirbs/etc/makefiles/tac_db_import.mk GSMA_HOME=/data/
  gsma_tac all
  ```
- To import operator data with operator ID operator_id:
  ```
  make -f /opt/dirbs/etc/makefiles/operator_import.mk
  OPERATOR_ID=operator_id all
  ```
- To import stolen_list data:
  ```
  make -f /opt/dirbs/etc/makefiles/stolen_list_import.mk
  STOLEN_LIST_HOME=/data/stolen_list all
  ```
- To import pairing_list data:
  ```
  make -f /opt/dirbs/etc/makefiles/pairing_list_import.mk
  PAIRING_LIST_HOME=/data/pairing_list all
  ```
- To import registration_list data:
  ```
  make -f /opt/dirbs/etc/makefiles/registration_list_import.mk
  REGISTRATION_LIST_HOME=/data/registration_list all
  ```
- To import golden_list data:
  ```
  make -f /opt/dirbs/etc/makefiles/golden_list_import.mk
  GOLDEN_LIST_HOME=/data/golden_list all
  ```

Jenkins

If you are using Jenkins to trigger the above Makefile invocations, Jenkins will not, by default, create a login shell, and the DIRBS Core virtualenv will not be activated. In this case, virtualenv activation must precede the call to `make`:

```
. /home/dirbs/dirbs-venv/bin/activate
```

This makes the total command, as run under Jenkins via an SSH slave, similar to:


```
. /home/dirbs/dirbs-venv/bin/activate && make -f /opt/dirbs/etc/makefiles/
stolen_list_import.mk STOLEN_LIST_HOME=/data/stolen_list all
```

3.5 Classification of IMEIs – `dirbs-classify`

The `dirbs-classify` command runs analysis on imported data, based on the configured conditions in the .yml configuration file. Analysis should be run prior to running `dirbs-listgen` and `dirbs-reports`.

A sample configuration for the conditions in this section is provided in Appendix B.

Table 3-14 Implemented dimensions and parameters lists the implemented dimensions and their parameters in release 16.0.0.

Table 3-14 Implemented dimensions and parameters

Asset	Function
gsma_not_found	Determines whether an IMEI is in the GSMA TAC database **Note:** Do not use this condition if there is a live DRS enforcing GSMA not found.
stolen_list	Matches IMEIs on the local stolen list
duplicate_threshold	▪ Matches duplicate IMEIs where the number of triplets (IMEI/IMSI/MSISDN combinations) with IMEI exceeds the threshold over configurable period ▪ Required parameters: ▫ threshold: Threshold of IMSIs that an IMEI must be seen with to be considered a duplicate (inclusive) ▫ period_days or period_months: Number of days or months in history to consider for duplicate analysis (only one of these can be specified) ▫ use_msisdn: To use MSISDN rather than IMSI for analysis the use_msisdn parameter should be True, by default it is False.
duplicate_daily_avg	▪ Matches duplicate IMEIs where the average daily number of IMSIs seen with that IMEI over a configurable period exceeds a configurable threshold if that IMEI was seen on at least a configurable number of days during that period ▪ Required parameters: ▫ threshold: Floating point number of daily average IMSIs that an IMEI must be seen with to be considered a duplicate (inclusive) ▫ period_days or period_months: Number of days or months in history to consider for duplicate analysis (only one of these can be specified) ▫ min_seen_days: Minimum number of days that an IMEI must be seen before it can be considered a duplicate (used to avoid averaging a small number of data points)

	▫ use_msisdn: To use MSISDN rather than IMSI for analysis the use_msisdn parameter should be True, by default it is False.
malformed_imei	▪ Matches IMEIs containing a non-digit character in the first 14 characters ▪ Matches IMEIs that are not 14 characters in length
not_on_registration_list	Matches IMEIs that do not appear on the registration list
inconsistent_rat	Matches IMEIs whose observed RAT on the network does not match model capabilities in GSMA TAC DB
used_by_dirbs_subscriber	▪ Matches IMEIs seen with an IMSI belonging to a configured DIRBS operator (MCC-MNC match) ▪ Can be used as part of a compound condition to specify different business rules when IMEI was seen with at least one local DIRBS subscriber ▪ Required parameters: ▫ lookback_days: Maximum number of days to look back when considering whether IMEI was seen with a DIRBS subscriber
used_by_international_roamer	▪ Matches IMEIs seen with an IMSI where the MCC did not match one of the configured MCCs for the DIRBS country ▪ Can be used as part of a compound condition to specify different business rules when IMEI was seen with at least one international roamer ▪ Required parameters: ▫ lookback_days: Maximum number of days to look back when considering whether IMEI was seen with an international roamer
used_by_local_non_dirbs_roamer	▪ Matches IMEIs seen with an IMSI belonging to the DIRBS country but not a configured MCC-MNC ▪ Intended to target an edge case where not all national operators might be analyzed DIRBS and/or where only certain regions were targeted ▪ Can be used as part of a compound condition to define different business rules for these cases ▪ Required parameters: ▫ lookback_days: Maximum number of days to look back when considering whether IMEI was seen with a DIRBS subscriber
is_test_tac	▪ Determines whether an IMEI belongs to a TAC that is classified as test TAC.
exists_in_barred_list	▪ Determines whether an IMEI is on the barred list. ▪ Intended to target the use case where the authority wants to blacklist or classify certain IMEIs irrespective of any reason or classified reason.
is_barred_tac	▪ Determines whether an IMEI belongs to a TAC that is on the barred tac list. ▪ Intended to target the use case where an authority wants to block an entire series of IMEIs belonging to a specific TAC.

daily_avg_uid	▪ Matches duplicate IMEIs where the average daily number of UIDs seen with an IMEI over a configurable period exceeds a configurable threshold if that IMEI was seen on at least a configurable number of days during that period. ▪ Required parameters: ▫ threshold: Floating point number of daily average IMSIs that an IMEI must be seen with to be considered a duplicate (inclusive) ▫ period_days or period_months: Number of days or months in history to consider for duplicate analysis (only one of these can be specified) ▫ min_seen_days: Minimum number of days that an IMEI must be seen before it can be considered a duplicate (used to avoid averaging a small number of data points)
exists_in_monitoring_list	▪ Matches IMEIs against the operator data dump which also exists in the Monitoring List ▪ Required parameters: ▫ monitored_days: Integer number of days an IMEI must be in monitoring list before getting classified, default value is 0
not_on_association_list	▪ Matches IMEIs against the operator data which does not exists in the Device Association List
transient_imei	▪ Matches transient IMEIs where the average number of sequential MSISDNs appeared on the same network, exceeds a configurable threshold over a configurable period of days. ▪ Required Parameters: ▫ num_msisdns: Integer value of number of average MSISDNs that an IMEI must be seen with, to be considered as transient ▫ period: number of days in history to consider for analysis

To run classification using the conditions specified in the config file, use:

```
dirbs-classify
```

For help on all the options available to `dirbs-classify`, run:

```
dirbs-classify --help
```

```
Usage: dirbs-classify [OPTIONS]

  DIRBS script to classify IMEIs.

  Iterates through all configured conditions and write to the
  classification_state table.

Options:
  --conditions TEXT               By default, dirbs-classify classifies on all
                                  conditions. Specify a comma-separated list
                                  of condition names if you wish to classify
                                  only on those conditions. The condition name
                                  corresponds to the label parameter of the
                                  condition in the DIRBS configuration file.
```



```
  --safety-check / --no-safety-check
                                  DANGEROUS: Disables safety check that
                                  ensures that no more than a certain ratio of
                                  IMEIs will be classified.
  --curr-date TEXT                DANGEROUS: Sets current date in YYYYMMDD
                                  format for testing. By default, uses system
                                  current date.
  --disable-sanity-checks         If set sanity checks on classification will be
                                  disabled.
  --version                       Show the version and exit.
  -v, --verbose                   Print debug console output - file output is
                                  unaffected.
  --db-password-prompt            If set, will prompt the user for a
                                  PostgreSQL password rather than reading from
                                  config.
  --db-user TEXT                  The PostgreSQL DB database user to connect
                                  as.
  --db-name TEXT                  The PostgreSQL DB database name to connect
                                  to.
  --db-port INTEGER               The PostgreSQL DB port to connect to.
  --db-host TEXT                  The PostgreSQL DB host to connect to.
  --statsd-prefix TEXT            The environment prefix to prepend to all
                                  StatsD metrics.
  --statsd-port INTEGER           The StatsD port to connect to on the
                                  configured host.
  --statsd-host TEXT              The StatsD host to send metrics to.
  --max-db-connections INTEGER    The maximum DB connections to use
                                  concurrently during this job.
   --max-local-cpus INTEGER       The maximum number of local CPUs to use
                                   concurrently during this job.
  --help                          Show this message and exit.
```

If a specific limited list of conditions must be run instead of all the conditions listed on the configuration file .dirbs.yml, the `--conditions` option can be used. Use the condition label to be run.

Example

```
dirbs-classify --conditions simple_dimension
```

where the condition `simple_dimension` is the label parameter of the condition in the DIRBS configuration or .dirbs.yml as shown below:

```
conditions:
  - label: simple_dimension
    dimensions:
      - module: gsma_not_found
    grace_period_days: 30
    blocking: true
    reason: Violated simple dimension
    max_allowed_matching_ratio: 0.1
```

This command classifies all the IMEIs and stores the results in the database for list generation. It can be trivially scheduled using a crontab or Jenkins job to allow for daily classification.

3.6 Generating lists – `dirbs-listgen`

List generation takes place after classification.

To run list generation, run:

```
dirbs-listgen <output_dir>
```

where `output_dir` is a directory where the various lists will be output. `dirbs-listgen` automatically creates a timestamp-based subdirectory under this directory. There is no need for this directory to be empty.

Running `listgen` with no explicit curr-date parameter will base the end of its lookback window off the most recent operator data date, rather than the current date.

Table 3-15 DIRBS Core lists lists the different types of lists created by `dirbs-listgen`.

Table 3-15 DIRBS Core lists

List	Function
blacklist	▪ Lists IMEIs which have met a blocking condition and where the current date has exceeded the block date. ▪ This list is distributed to all operators and is the same for each.
notification lists	▪ Lists IMEIs which have met a blocking condition where the current date is still within the grace period for the condition. ▫ Does not include any IMEI already on the blacklist. ▪ For each IMEI, we generate subscriber triplets based on imported operator data. There is one row in the list for each triplet. ▪ For each triplet, we determine who the home network is based on the IMSI and the configured MCC/MNC pairs for each configured operator. ▪ If a triplet does not match any MCC/MNC pairing for a configured operator (roamers, etc.), we notify all operators whose data they have been seen in. ▪ Each operator gets a different list containing their subscribers and any fallback triplets seen on their network.
exception lists	▪ Each operator gets a copy of the pairing list, split into per-operator exception lists based again on their IMSI and the configured MCC/MNC pairs for the configured operators. ▪ If a pairing's IMSI does match any MCC/MNC pairing for a configured operator (roamers, etc.), the pairing is placed on every MNO exception list.

For help on all the options available to `dirbs-listgen`, run:

```
dirbs-listgen --help
```

```
Usage: dirbs-listgen [OPTIONS] OUTPUT_DIR

  DIRBS script to output CSV lists (blacklist, exception, notification) for
  the current classification state.

Options:
  --version                     Show the version and exit.
  -v, --verbose                 Print debug console output - file output is
                                unaffected.
  --db-password-prompt          If set, will prompt the user for a PostgreSQL
                                password rather than reading from config.
```



```
  --db-user TEXT                  The PostgreSQL DB database user to connect as.
  --db-name TEXT                  The PostgreSQL DB database name to connect to.
  --db-port INTEGER               The PostgreSQL DB port to connect to.
  --db-host TEXT                  The PostgreSQL DB host to connect to.
  --statsd-prefix TEXT            The environment prefix to prepend to all
                                  StatsD metrics.
  --statsd-port INTEGER           The StatsD port to connect to on the
                                  configured host.
  --statsd-host TEXT              The StatsD host to send metrics to.
  --max-db-connections INTEGER    The maximum DB connections to use concurrently
                                  during this job.
  --max-local-cpus INTEGER        The maximum number of local CPUs to use
                                  concurrently during this job.
  --curr-date TEXT                Sets current date in YYYYMMDD format for
                                  testing. By default, uses system current date.
  --disable-sanity-checks         If set sanity checks on classification will be
                                  disabled.
  --no-full-lists                 If set, disable outputting full lists as CSV
                                  for a performance improvement.
  --no-cleanup                    If set, intermediate tables used to calculate
                                  lists will not be deleted so that they can be
                                  inspected.
  --base INTEGER                  If set, will use this run ID as the base for
                                  the delta CSV lists.
  --help                          Show this message and exit.
```

The `dirbs-listgen` command can be trivially scheduled using a crontab or Jenkins job to allow for daily list generation.

To generate lists for specific conditions instead of all conditions configured in the config.yml file, use --conditions option.

3.7 Generating DIRBS reports – `dirbs-report`

Several reports can be generated using the `dirbs-report` command. The `dirbs-report report_type` command generates all reports in DIRBS Core (see Table 3-16 Report types).

Note: DIRBS Core reports will be depreciated as all the reporting will be implemented in DIRBS View.

Table 3-16 Report types

Report commands (types)	Function
condition_imei_overlaps	Generates per-condition reports showing matched IMEIs seen on more than one MNO network
gsma_not_found	Generates report of all GSMA not found IMEIs as CSV
standard	Generates standard monthly per-operator and country-level reports as HTML, CSV and JSON
stolen_violations	Generates per-MNO list of IMEIs seen on the network after they were reported stolen.
top_duplicates	Generates report listing IMEIs seen with more than 5 IMSIs in a given month and year as CSV
non_active_pairs	Generates report listing IMEIs, IMSIs pairs which are not active over the network for a specified period

blacklist_violation	Generates per-operator report of blacklist violations
association_list_violations	Generates per-operator association-list violations
classified_triplets	Generates list of classified triplets (IMEI, IMSI, MSISDN) for a specified condition
transient_msisdns	Generates list of transient MSISDNs where average number of sequential IMEIs it is observed with, exceeds a specified threshold over a specified analysis period.

For help on all the options available to `dirbs-report`, run:

```
dirbs-report --help
```

```
Usage: dirbs-report [OPTIONS] COMMAND [ARGS]...

  DIRBS script to output reports (operator and country) for a given MONTH
  and YEAR.

Options:
  --version              Show the version and exit.
  -v, --verbose          Print debug console output - file output is
                         unaffected.
  --db-password-prompt   If set, will prompt the user for a PostgreSQL
                         password rather than reading from config.
  --db-user TEXT         The PostgreSQL DB database user to connect as.
  --db-name TEXT         The PostgreSQL DB database name to connect to.
  --db-port INTEGER      The PostgreSQL DB port to connect to.
  --db-host TEXT         The PostgreSQL DB host to connect to.
  --statsd-prefix TEXT   The environment prefix to prepend to all StatsD
                         metrics.
  --statsd-port INTEGER  The StatsD port to connect to on the configured host.
  --statsd-host TEXT     The StatsD host to send metrics to.
  --help                 Show this message and exit.

Commands:
  condition_imei_overlaps  Generate per-condition reports showing...
  gsma_not_found           Generate report of all GSMA not found IMEIs.
  standard                 Generate standard monthly operator and...
  stolen_violations        Generate per-MNO list of IMEIs seen on the...
  top_duplicates           Generate report listing IMEIs seen with more...
```

```
dirbs-report condition_imei_overlaps --help
```

```
Usage: dirbs-report condition_imei_overlaps [OPTIONS] MONTH YEAR OUTPUT_DIR

  Generate per-condition reports showing matched IMEIs seen on more than one
  MNO network.

Options:
  --max-db-writers INTEGER        The maximum write-intensive DB connections
                                  to use concurrently during this job.
  --max-db-connections INTEGER    The maximum DB connections to use
                                  concurrently during this job.
  --max-local-cpus INTEGER        The maximum number of local CPUs to use
                                  concurrently during this job.
  --debug-query-performance       Enable this to print out more stats about
                                  duration of queries during stats generation.
  --disable-data-check            Disable check to validate existence of data
                                  for all configured operators in this
                                  reporting month.
  --disable-retention-check       Disable check that stops reports being run
                                  for months outside the retention period.
  --force-refresh / --no-refresh  Whether data in report should be refreshed
                                  from latest data or from previously-
```



```
                                calculated data (default: --no-refresh).
  --help                        Show this message and exit.
```

dirbs-report gsma_not_found --help

```
Usage: dirbs-report gsma_not_found [OPTIONS] MONTH YEAR OUTPUT_DIR

  Generate report of all GSMA not found IMEIs.

Options:
  --max-db-writers INTEGER        The maximum write-intensive DB connections
                                  to use concurrently during this job.
  --max-db-connections INTEGER    The maximum DB connections to use
                                  concurrently during this job.
  --max-local-cpus INTEGER        The maximum number of local CPUs to use
                                  concurrently during this job.
  --debug-query-performance       Enable this to print out more stats about
                                  duration of queries during stats generation.
  --disable-data-check            Disable check to validate existence of data
                                  for all configured operators in this
                                  reporting month.
  --disable-retention-check       Disable check that stops reports being run
                                  for months outside the retention period.
  --force-refresh / --no-refresh  Whether data in report should be refreshed
                                  from latest data or from previously-
                                  calculated data (default: --no-refresh).
  --help                          Show this message and exit.
```

dirbs-report standard --help

```
Usage: dirbs-report standard [OPTIONS] MONTH YEAR OUTPUT_DIR

  Generate standard monthly operator and country-level reports.

Options:
  --max-db-writers INTEGER        The maximum write-intensive DB connections
                                  to use concurrently during this job.
  --max-db-connections INTEGER    The maximum DB connections to use
                                  concurrently during this job.
  --max-local-cpus INTEGER        The maximum number of local CPUs to use
                                  concurrently during this job.
  --debug-query-performance       Enable this to print out more stats about
                                  duration of queries during stats generation.
  --disable-data-check            Disable check to validate existence of data
                                  for all configured operators in this
                                  reporting month.
  --disable-retention-check       Disable check that stops reports being run
                                  for months outside the retention period.
  --force-refresh / --no-refresh  Whether data in report should be refreshed
                                  from latest data or from previously-
                                  calculated data (default: --no-refresh).
  --help                          Show this message and exit.
```

dirbs-report stolen_violations --help

```
Usage: dirbs-report stolen_violations [OPTIONS] OUTPUT_DIR

  Generate per-MNO list of IMEIs seen on the network after they were
  reported stolen.

Options:
  --max-db-writers INTEGER      The maximum write-intensive DB connections to
                                use concurrently during this job.
```



```
  --max-db-connections INTEGER  The maximum DB connections to use concurrently
                                during this job.
  --max-local-cpus INTEGER      The maximum number of local CPUs to use
                                concurrently during this job.
  --newer-than TEXT             Include violations newer than the date passed
                                in YYYYMMDD format.
  --filter-by-conditions TEXT   Specify a comma-separated list of condition
                                names if you wish to filter by those
                                conditions.
  --help                        Show this message and exit.
```

```
dirbs-report top_duplicates --help
```

```
Usage: dirbs-report top_duplicates [OPTIONS] MONTH YEAR OUTPUT_DIR

  Generate report listing IMEIs seen with more than 5 IMSIs in a given month
  and year.

Options:
  --max-db-writers INTEGER        The maximum write-intensive DB connections
                                  to use concurrently during this job.
  --max-db-connections INTEGER    The maximum DB connections to use
                                  concurrently during this job.
  --max-local-cpus INTEGER        The maximum number of local CPUs to use
                                  concurrently during this job.
  --debug-query-performance       Enable this to print out more stats about
                                  duration of queries during stats generation.
  --disable-data-check            Disable check to validate existence of data
                                  for all configured operators in this
                                  reporting month.
  --disable-retention-check       Disable check that stops reports being run
                                  for months outside the retention period.
  --force-refresh / --no-refresh  Whether data in report should be refreshed
                                  from latest data or from previously-
                                  calculated data (default: --no-refresh).
  --help                          Show this message and exit.
```

`output_dir` is the existing directory where HTML, JS, CSS, CSV, and JSON files will be output. `dirbs-report` automatically creates a timestamp-based subdirectory under this directory so there is no need for this directory to be empty.

3.7.1 `dirbs-report` directory structure

Generated output from the `dirbs-report` command will be placed in the specified `output_dir`.

The `output_dir` will contain the HTML, JS, CSS, CSV, and JSON files, and based on the following directory naming convention:
`'report'_'subcommand'_'timestamp'_'run_id'_'class_run_id'_'data_id'_'month'_'year'`

where:

- `subcommand` is the `dirbs-report` subcommand
  - standard, gsma_not_found, top_duplicates, condition_imei_overlaps, stolen_violations
- `timestamp` is the run_id_start_time in the job_metadata table

- Format is %Y%m%d_%H%M%S, i.e., 20171102_051731

- `run_id` increments each time a report is run, i.e., 'run_id_4
- `class_run_id` is the classification id of the last successful `dirb-classify` run, i.e., 'class_id_3'
- `data_id`, i.e., data_id_1
- `month`, i.e., month_7
- `year`, i.e., year_2016

Sample listing of directory names for various subcommands

- standard

  `report_standard_20171102_052206_run_id_5_class_id_3_data_id_1_month_7_year_2016`

- condition_imei_overlaps

  This is the same name structure as gsma_not_found and top_duplicate subcommands. data_id is not used for these subcommands.

  `report_condition_imei_overlaps_20171102_052800_run_id_6_class_id_3_month_11_year_2016`

- stolen_violation_directory

  Month, year and data_id are not used for this subcommand

  `report_stolen_violations_20171102_051731_run_id_4_class_id_3`

E: Visual reports depend on the JSON data, so it is not possible to publish just the HTML, CSS and JS files. Due to security restrictions imposed by the browser, HTML files generated by `dirbs-report` must be hosted by a webserver rather than opened locally from the filesystem. If you open the reports from the file system, you will receive an alert box stating that the JSON data could not be loaded.

3.8 Accessing the API server

The API server provides information on the data catalog, job metadata, TAC, IMEI, MSISDN, and DIRBS code and schema version

Assuming that you have published the container's port 5000 to the host using the `-p 5000:5000` option to `docker run`, you should be able to open a web browser on the host machine and access the API server on the:

- Data catalog API Version 1.0 (see Section 3.8.1)
- Data catalog API Version 2.0 (see Section 3.8.2)
- Job metadata API Version 1.0 (see Section 3.8.3)
- Job metadata API Version 2.0 (see Section 3.8.4)
- TAC API Version 1.0 (see Section 3.8.5)
- TAC API Version 2.0 (see Section 3.8.6)

- IMEI API Version 1.0 (see Section 3.8.7)
- IMEI API Version 2.0 (see Section 3.8.8)
- MSISDN API Version 1.0 (see Section 3.8.9)
- MSISDN API Version 2.0 (see Section 3.8.10)
- Version API Version 1.0 (see Section 3.8.11)
- Version API Version 2.0 (see Section 3.8.12)

3.8.1 Data catalog API (Version 1.0)

Table 3-17 Data catalog API

API endpoint	Description
/api/v1/catalog	Returns last 100 entries from the data_catalog table sorted by last_seen timestamp in descending order
/api/v1/catalog?max_results=1	▪ Returns last 'x' entries from the data_catalog table sorted by last_seen timestamp in descending order ▫ 'x' is specified in max_results parameter
/api/v1/catalog?file_type=gsma_tac	▪ Returns last 100 entries of file_type 'x' from the data_catalog table sorted by last_seen timestamp in descending order ▫ 'x' is specified by file_type parameter
/api/v1/catalog?is_valid_zip=True	▪ Returns last 100 entries with is_valid_zip status equal to 'x' from the data_catalog table sorted by last_seen timestamp in descending order ▫ 'x' is specified in is_valid_zip parameter
/api/v1/catalog?modified_since=20170825	▪ Returns last 100 entries with modified_time greater than equal to 'x' from the data_catalog table sorted by last_seen timestamp in descending order ▫ 'x' is specified in modified_since parameter
/api/v1/catalog?cataloged_since=20170801	▪ Returns last 100 entries with last_seen greater than equal to 'x' from the data_catalog table sorted by last_seen timestamp in descending order ▫ 'x' is specified in cataloged_since parameter
/api/v1/catalog?modified_since=20170825&is_valid_zip=True	▪ Returns last 100 entries with last_seen greater than equal to 'x' and is_valid_zip equal to 'y' from the data_catalog table sorted by last_seen timestamp in descending order ▫ 'x' is specified in cataloged_since and 'y' is_valid_zip parameters

3.8.2 Data catalog API (Version 2.0)

The Data Catalog API Version 2.0 is the extended version of API Version 1.0, it includes some extra features (having the previous features as well) which are described in the following:

- **Pagination support** has been introduced in it, now results can be paginated using the keys **limit** and **offset**

  e.g `api/v2/catalog?limit=10&offset=1`

  Now this will show total number of 10 results starting from the first. More details can be views in API Specification Guidelines.

- Result **sorting** support has been introduced in ascending or descending order, now results can be sorted based on the **file_id** of the file in ascending or descending order.

  e.g. `api/v2/catalog?limit=10&offset=1&order=Ascending`

  `api/v2/catalog?limit=10&offset=1&order=Descending`

  Now these will sort the results in ascending or descending orders based on the id of the file cataloged.

3.8.3 Job metadata API (Version 1.0)

Table 3-18 Job metadata API

API endpoint	Description
/api/v1/job_metadata?max_results=<n> (n defaults to 10)	The number of jobs to show in this list can be configured by the <max_results> query parameter, which defaults to 10 and **must be a positive integer**.
/api/v1/job_metadata?run_id=<first_run_id>&run_id=<second_run_id> (defaults to any run_id)	Jobs can be filtered by a list of run_ids using the <run_id> query parameter, which defaults to any run_id. Each run_id **must be a positive integer**.
/api/v1/job_metadata?command=<first_command_name_without_quotes>&command=<second_command_name_without_quotes>(defaults to any command) e.g. for import job: /api/v1/job_metadata?command=dirbs-import&command=dirbs-prune	Job command name can be specified using the <command> query parameter, which defaults to any command. It is possible to specify more than one command name in the same query using the symbol "&" to concatenate the params. **Each command name must refer to an existing command**, such as:

	"dirbs-import", "dirbs-classify", "dirbs-prune", "dirbs-listgen", "dirbs-catalog", "dirbs-report", "dirbs-db" .
/api/v1/job_metadata?subcommand=<subcommand_name> (defaults to any sub_command)	Jobs can be filtered by a list of job subcommands using the <subcommand> query parameter, which defaults to any subcommand.
/api/v1/job_metadata?status=error&status=success (defaults to any status)	Jobs can be filtered by a list of job metadata using the <status> query parameter, which defaults to any status. Each job metadata **must be either 'running', 'success' or 'error'**.
/api/v1/job_metadata?show_details=True (defaults to True)	Extra details for the specific job can be retrieved in the extra_metadata section in the JSON response by setting <show_details> query parameter to True, which is the default value. Show_details **must have a boolean value: True, False, 0, 1.** If show_details is set to False, extra_metadata section will not be included in the JSON response.
/api/v1/job_metadata?show_details=True&status=error&status=success&max_results=3&command=dirbs-import&command=dirbs-prune	Query parameters for jobs can be repeated to allow multiple values for the same param (if eligible). All query params are eligible for multiple values except max_results and show_details. Multiple filters can be combined in the same query by adding query parameters separated by the symbol '&' (first query param must start with symbol '?').

3.8.4 Job Metadata API (Version 2.0)

The Job Metadata API Version 2.0 is the extended version of API Version 1.0, it includes some extra features (having the previous features as well) which are described in the following:

- **Pagination support** has been introduced in it, now results can be paginated using the keys **limit** and **offset**

  e.g `api/v2/job_metadata?limit=10&offset=1`

  Now this will show total number of 10 results starting from the first. More details can be views in API Specification Guidelines.

- Result **sorting** support has been introduced in ascending or descending order, now results can be sorted based on the **run_id** of the job in ascending or descending order.

e.g. `api/v2/job_metadata?limit=10&offset=1&order=Ascending`

`api/v2/job_metadata?limit=10&offset=1&order=Descending`

Now these will sort the results in ascending or descending orders based on the run id of the job.

3.8.5 TAC API (Version 1.0)

The TAC API returns relevant data from the GSMA TAC DB. The GSMA TAC fields for NFC, Bluetooth and WLAN are displayed as the raw content from the GSMA TAC DB.

`http://localhost:5000/api/v1/tac/<tac_num>`

where `<tac_num>` is the 8-digit TAC.

3.8.6 TAC API (Version 2.0)

The API Version 2.0 supports both **GET** and **POST** methods in this version for the TAC API. The GET method returns the same relevant data from GSMA TAC DB as in the previous version however the TAC POST API (Batch TAC API) accepts 1000 TACs at the same time and returns results for them.

e.g. **`curl -X POST --header "Content-Type: application/json" –header "Accept: application/json" -d "{`**

```
\"tacs\": [
            \"12345678\",
            \"56478377\"
          ]
}" "http://localhost:5000/api/v2/tac"
```

Now this will return results for both TACs from GSMA TAC Database. More details can be viewed in API Specifications Guidelines.

3.8.7 IMEI API (Version 1.0)

The IMEI API returns all known information about the IMEI, as well as results of all 'conditions' evaluated as part of DIRBS Core.

The following real-time checks are also included information:

- Invalid IMEI
- GSMA not found
- Registration status
- IMEIs ever observed on the network

`http://localhost:5000/api/v1/imei/<imei>?include_seen_with=<0,1,true, false>`

where

`include_seen_with` determines whether or not the `seen_with` field will be present in the response.
If the `include_seen_with` parameter is not set, it defaults to 0, meaning no `seen_with` data will be calculated or sent.

3.8.8 IMEI API (Version 2.0)

The IMEI API Version 2.0 has been divided into five different APIs which are as follow:

1. **IMEI API:** Returns information about an IMEI as the result of off conditions evaluated as a part of DIRBS Core. It also shows block date, real-time checks such as IMEI observation on network, its pairing status, if it is exempted device or if it is invalid. The IMEI API has two additional blocks in this version which are **registration_status** which shows IMEI status from Registration List and **stolen_status** which shows IMEI status from Stolen List.

   It can be accessed as: `api/v2/imei/<imei>`

2. **IMEI Info API:** Returns information about an IMEI's Brand Name, Device Type, Make, Model, Model Number, Radio Interfaces and current Status of it in the DIRBS-Core.

   It can be accessed as: `api/v2/imei/<imei>/info`

3. **IMEI Pairings API:** Returns information core knows about the IMSI's paired with the IMEI in the device pairing system. It supports pagination as well. It also supports results sorting in ascending or descending order based on MSISDN as key.

   It can be accessed using: `api/v2/imei/<imei>/pairings`

   It accepts **limit**, **offset** as pagination parameters and **Ascending** or **Descending** as sorting orders.

4. **IMEI Subscribers API:** Returns information Core knows about the IMSI-MSISDN pairs the IMEI has been seen with on the network. It supports pagination and sorting as well.

   It can be accessed using: `api/v2/imei/<imei>/subscribers`

   It accepts **limit**, **offset** as pagination parameters and **Ascending** or **Descending** as sorting orders.

5. **IMEI Batch API:** Returns information about IMEIs same as in single IMEI API described above. It accepts maximum 1000 IMEI's at the same time and return results for each one collectively.

   It can be accessed using POST: `api/v2/imei-batch`

3.8.9 MSISDN API (Version 1.0)

The MSISDN API returns a list of IMEI, IMSI, GSMA Manufacturer, GSMA Model Name for the MSISDN specified:
`http://localhost:5000/api/v1/msisdn/<msisdn>`

3.8.10 MSISDN API (Version 2.0)

The MSISDN API returns information about the MSISDN, such as IMEI, IMSI, GSMA Manufacturer, Model Name and Last Seen date for the specified MSISDN. It also returns Information from GSMA TAC Database as well as Device Registration System for the specified MSISDN.

It can be accessed using: `api/v2/msisdn/<msisdn>`

3.8.11 Version API (Version 1.0)

This simple API returns the code, DB schema version, report schema version and version of the source code of DIRBS Core.

It can be accessed as: `http://localhost:5000/api/v1/version`

3.8.12 Version API (Version 2.0)

This simple API returns the code, DB schema version, report schema version and version of the source code of DIRBS Core.

It can be accessed as: `http://localhost:5000/api/v1/version`

3.9 Pruning old data

Table 3-19 lists commands used to prune obsolete data from the DIRBS Core PostgreSQL.

Table 3-19 Prune commands

Prune commands	Function
blacklist	Expire IMEIs outside the blacklist retention period.
classification_state	Prune obsolete classification_state data.
lists	Prune obsolete list tables data.
triplets	Prune old seen_triplets data.

For help on all options available to `dirbs-prune`, run:

```
dirbs-prune --help
```

```
Usage: dirbs-prune [OPTIONS] COMMAND [ARGS]...

  DIRBS script to prune obsolete data from the DIRBS Core PostgreSQL
  database.

Options:
  --version               Show the version and exit.
  -v, --verbose           Print debug console output - file output is
                          unaffected.
  --db-password-prompt    If set, will prompt the user for a PostgreSQL
                          password rather than reading from config.
  --db-user TEXT          The PostgreSQL DB database user to connect as.
  --db-name TEXT          The PostgreSQL DB database name to connect to.
  --db-port INTEGER       The PostgreSQL DB port to connect to.
```



```
  --db-host TEXT          The PostgreSQL DB host to connect to.
  --statsd-prefix TEXT    The environment prefix to prepend to all StatsD
                          metrics.
  --statsd-port INTEGER   The StatsD port to connect to on the configured
host.
  --statsd-host TEXT      The StatsD host to send metrics to.
  --curr-date TEXT        Sets current date in YYYYMMDD format for
testing. By
                          default, uses system current date.
  --help                  Show this message and exit.

Commands:
  blacklist              Expire IMEIs outside the blacklist retention...
  classification_state   Prune obsolete classification_state data.
  lists                  Prune obsolete lists data.
  triplets               Prune old seen_triplets data.
```

dirbs-prune classification_state --help

```
Usage: dirbs-prune classification_state [OPTIONS]

  Prune obsolete classification_state data.

Options:
  --help  Show this message and exit.
```

dirbs-prune blacklist --help

```
Usage: dirbs-prune blacklist [OPTIONS] [CONDITION_NAME]

  Expire IMEIs outside the blacklist retention period from blacklist.

Options:
  --prune-all  DANGEROUS: If set, will set end_date to all the imeis
          falling in the specified period
  --help  Show this message and exit.
```

dirbs-prune lists --help

```
Usage: dirbs-prune lists [OPTIONS]

  Prune obsolete lists data.

Options:
  --help  Show this message and exit.
```

dirbs-prune triplets --help

```
Usage: dirbs-prune triplets [OPTIONS]

  Prune old seen_triplets data.

Options:
  --help  Show this message and exit.
```


4 Understanding DIRBS Reports

4.1 Standard reports

Note: DIRBS Core reports will be depreciated as all the reporting is done in DIRBS View.

Standard monthly operator and country-level reports are generated in HTML, JSON and CSV formats. The formats and sections for the country and operator reports are the same. Operator reports are specific to their respective operators configured in the .dirbs.yml file. Country-level reports reflect all the IMEIs seen in the country.

The JSON file has a report schema version associated with any generated standard report and are explicit fields called “report_schema_version” and “software_version”

Expect to see the version number incremented when:

- Fields are added, removed or renamed
- The method of calculation for a field is changed so that it cannot be compared to previous reports

The standard report has a `--force-refresh / --no-refresh` (default) CLI option:

- `--no-refresh` reports can be generated very quickly since no numbers are calculated
- `--force-refresh` tells `dirbs-report` to re-do stats generation if there is previous data available for the same month and year
- Standard report only looks for previous data with the same report_schema_version
  - If schema has changed, `dirbs-report` will always generate new data

Placeholder reports were created with no data for configured operators that have no data for the month. These reports are only created when the CLI option `--disable-data-check` is used.

Other CLI options for placeholder reports are:

- `--max-db-connections <int>`: Determines the number of parallel jobs to perform during stats generation (performance scales linearly with this number).
- `--disable-data-check`: By default, `dirbs-report` ensures that there is data available for all operators for the given month and year before generating a report. Disabling this allows a report to be generated even if data for one operator’s data is missing.
- `--disable-retention-check`: By default, `dirbs-report` will fail if there is an attempt to generate a report outside the retention period.
- `--debug-query-performance`: Provides more detail in the console output about query performance during the stats generation phase.

4.1.1 Country report

4.1.1.1 HTML

The HTML country report covers:

- Identifier counts
- Identifier trends
- Compliance breakdown
- IMEI compliance trends
- Conditions breakdown
- Condition combinations
- Blacklist and blacklist violations
- Top models: counts
- Top models: gross adds

E:Figures in this section show graphic representations of the same sections in the JSON report.

Figure 4-1 Country report main page – HTML shows the main page for country reports in HTML. Navigate to different sections of the report by clicking on the navigation pane on the left.

Figure 4-1 Country report main page – HTML

Identifier counts

Identifier counts show a distinct number of:

- IMEIs, MSISDNs, and IMSIs
- Combination pairs of IMEI-IMSI, IMEI-MSISDN, and IMSI-MSISDN

- Triplet combinations of IMEI-IMSI-MSISDN

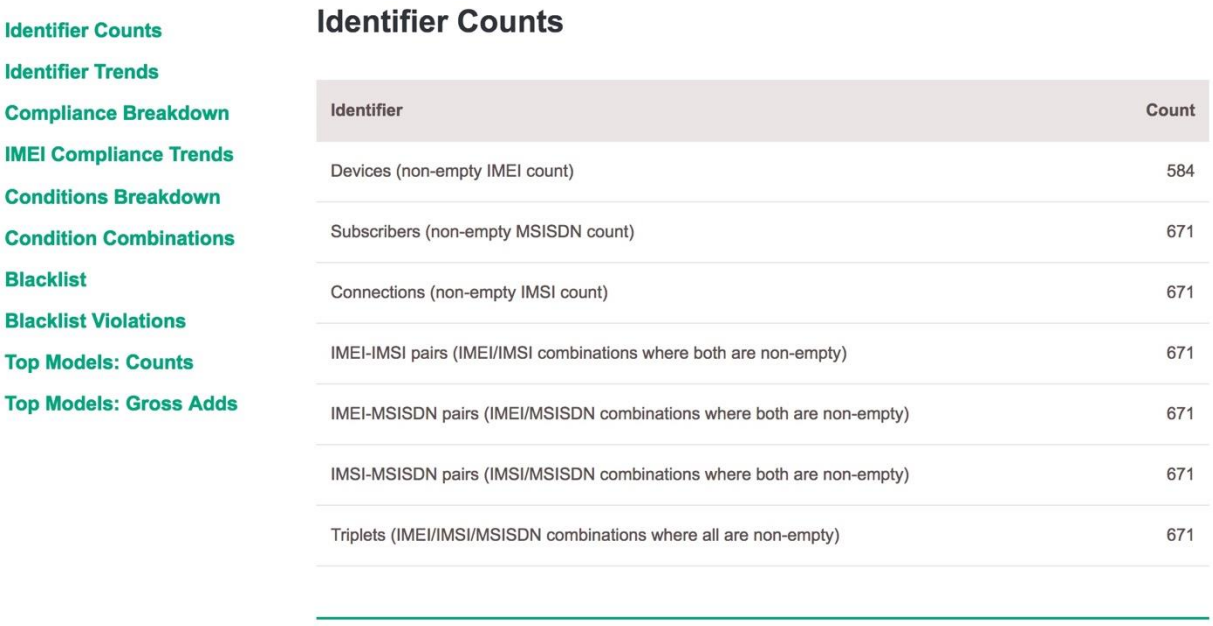


Figure 4-2 Identifier counts

Identifier trends

IMEIs, MSISDNs, and IMSIs counts for the months with data for the period specified in the configuration file.

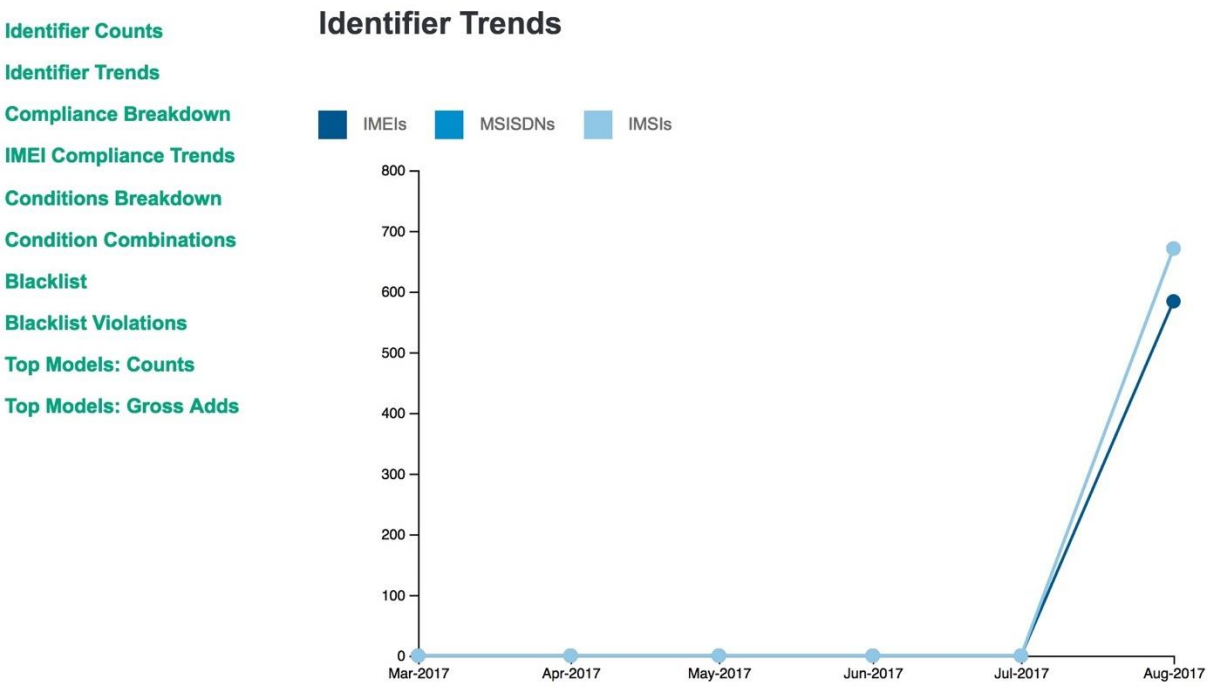
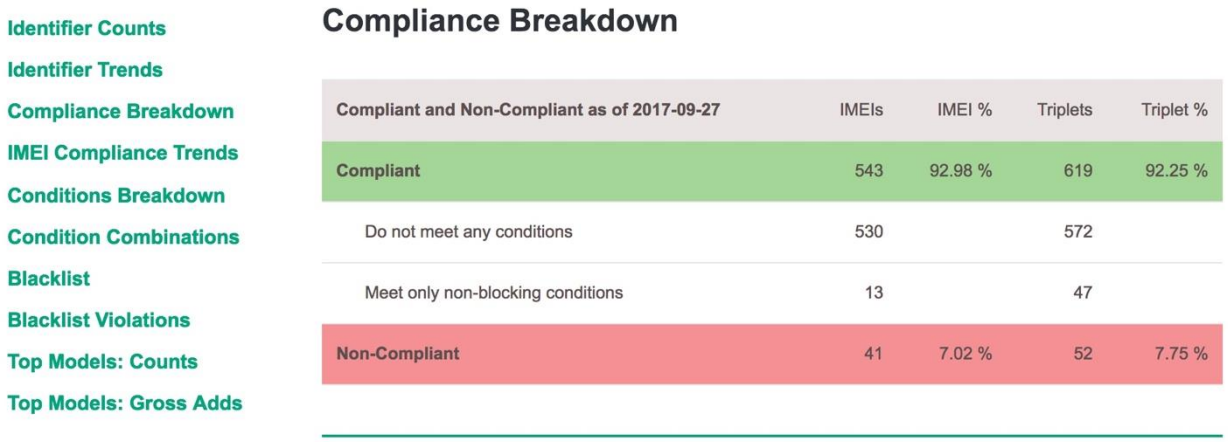


Figure 4-3 Identifier trends

Compliance breakdown

The compliance breakdown shows:

- Compliant IMEIs and triplets that do not meet any conditions or conditions that are non-blocking
- Non-compliant IMEIs and triplets that meet one or more blocking conditions



Compliant and Non-Compliant as of 2017-09-27	IMEIs	IMEI %	Triplets	Triplet %
Compliant	543	92.98 %	619	92.25 %
Do not meet any conditions	530		572	
Meet only non-blocking conditions	13		47	
Non-Compliant	41	7.02 %	52	7.75 %

Figure 4-4 Compliance breakdown

IMEI compliance trends

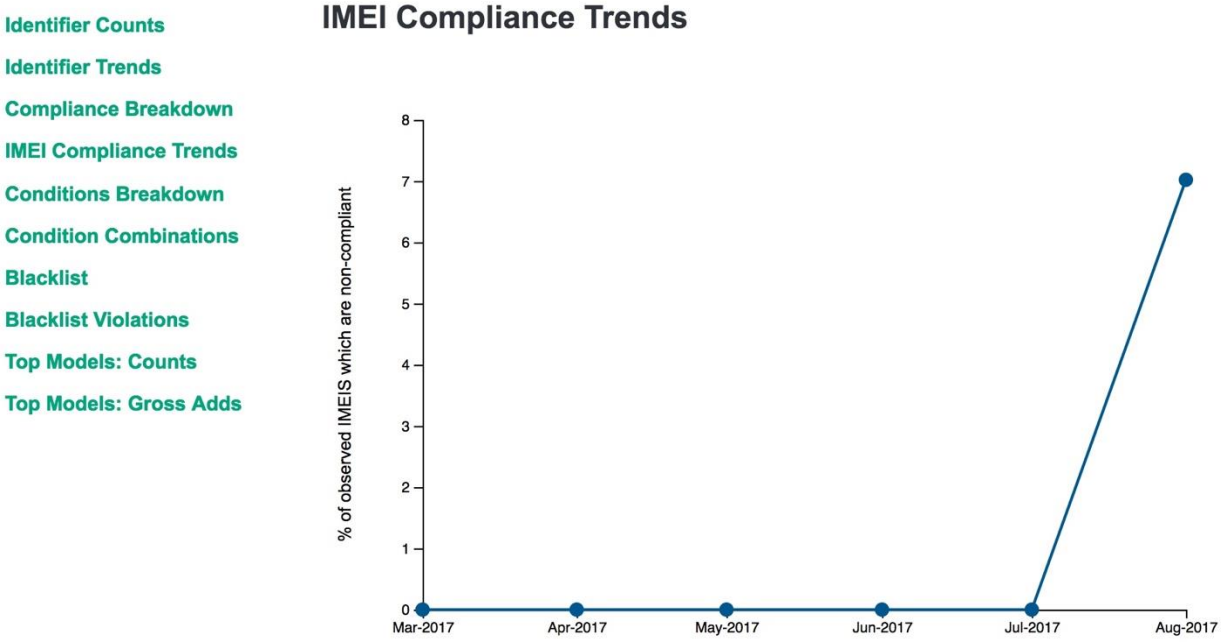
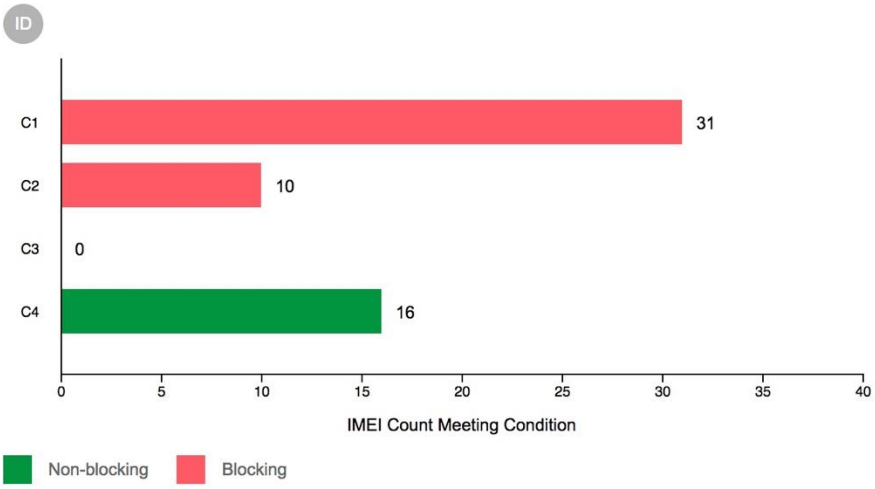


Figure 4-5 IMEI compliance trends

Conditions breakdown

Each condition is independent of each other. An IMEI can meet one or more conditions, and is counted on each condition it meets. The sum of the IMEIs for the conditions breakdown does not equal the number of IMEIs found on the compliance breakdown.

Conditions Breakdown



ID	Condition Name	Blocking	IMEIs	IMEI %	IMEI Trend
C1	gsma_not_found	Blocking	31	5.31 %	

Other counts and percentages

Gross Add IMEIs: 31 (5.31 %)
IMEI-MSISDN pairs: 39 (5.81 %)
IMEI-IMSI pairs: 39 (5.81 %)
IMEI-IMSI-MSISDN triplets: 39 (5.81 %)

Configuration (for last successful classification)

Dimensions:

- gsma_not_found

Grace period: 90 days

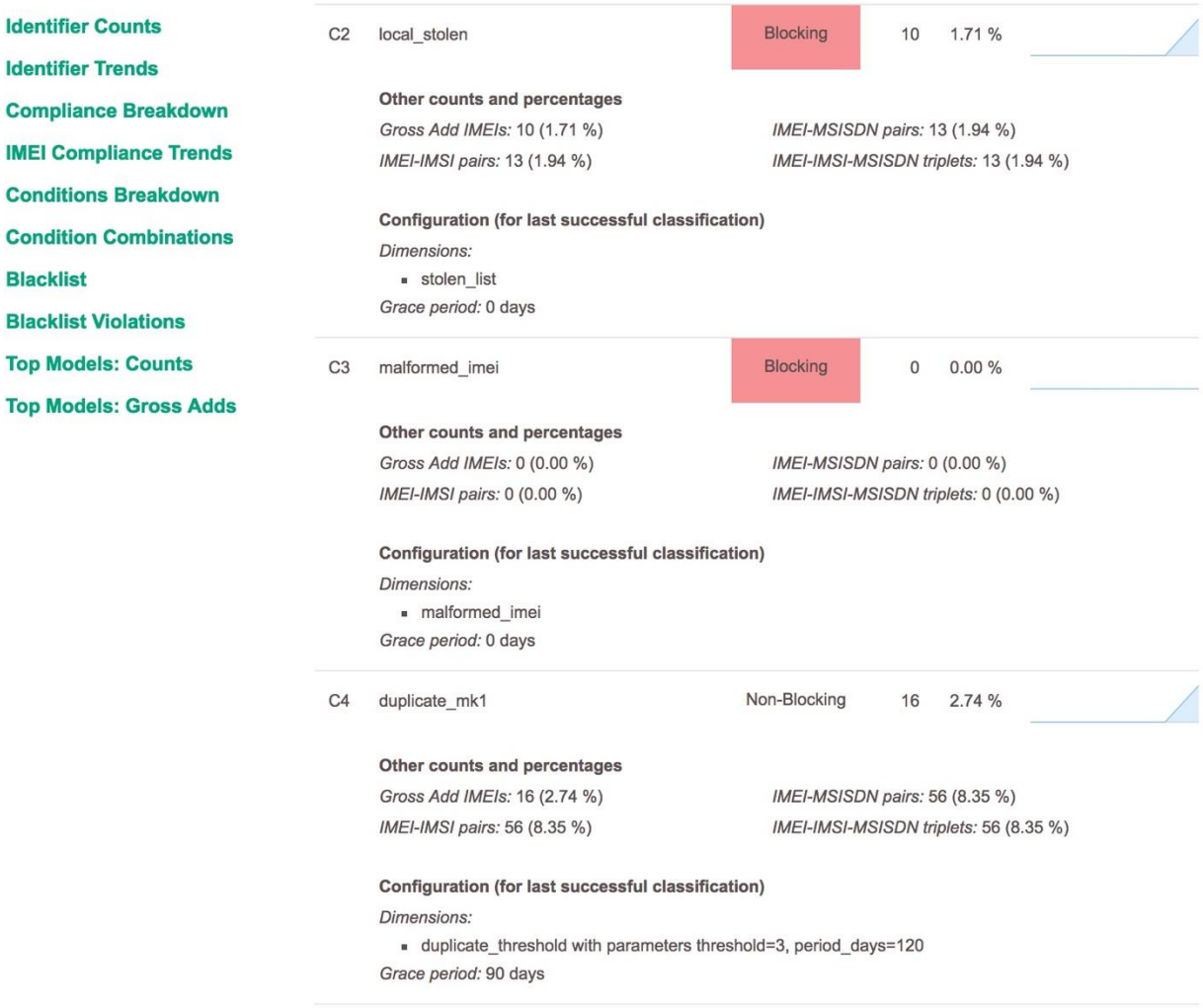


Figure 4-6 Conditions breakdown

Condition combinations

IMEIs, IMEI-IMSIs pairs, and triplets count for the conditions they meet including the combination of conditions.

Figure 4-7 Condition combinations

Blacklist and blacklist violations

Blacklists and blacklist violations report the number of blacklisted IMEIs.

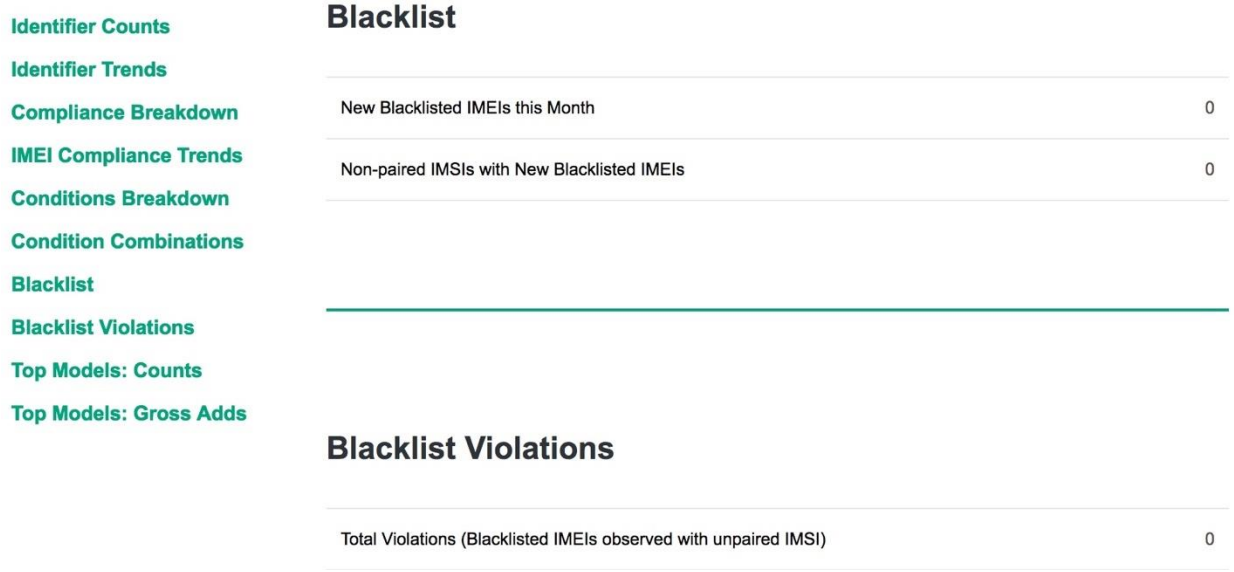


Figure 4-8 Blacklist and blacklist violations

Top models: counts

Top models show the top 10 models by IMEI counts.

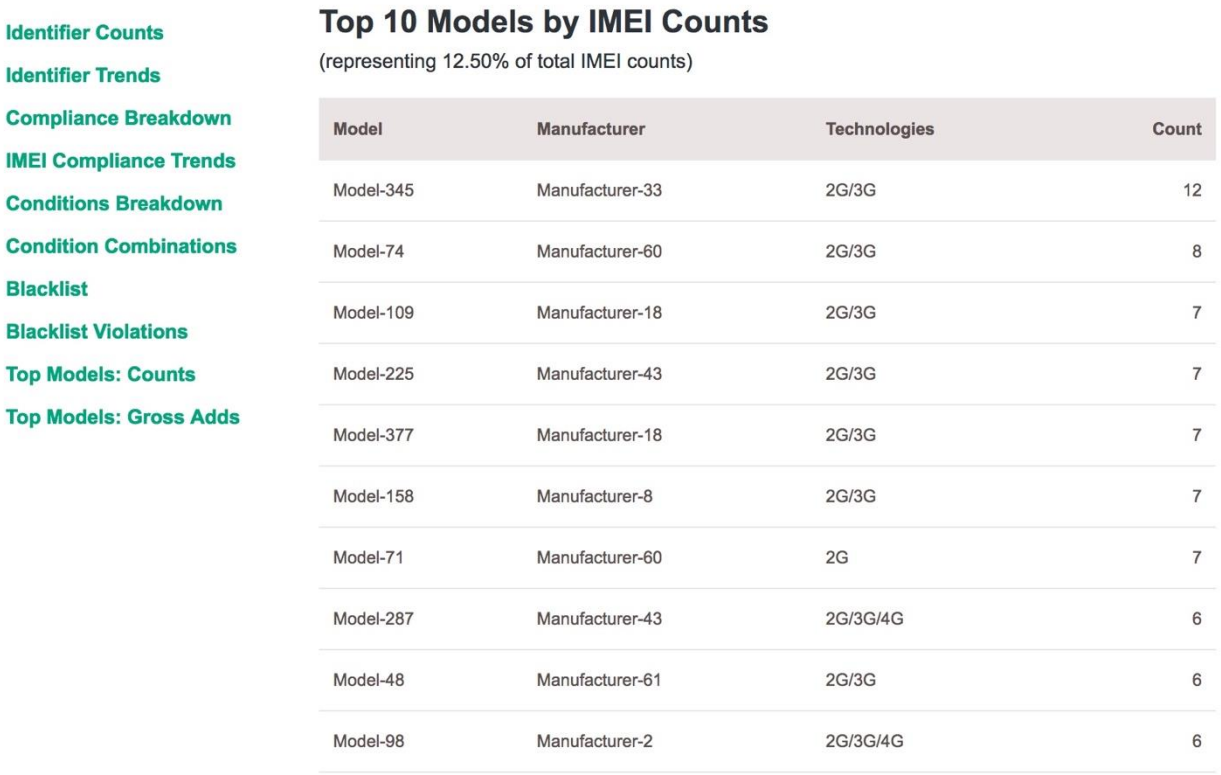


Figure 4-9 Top models: counts

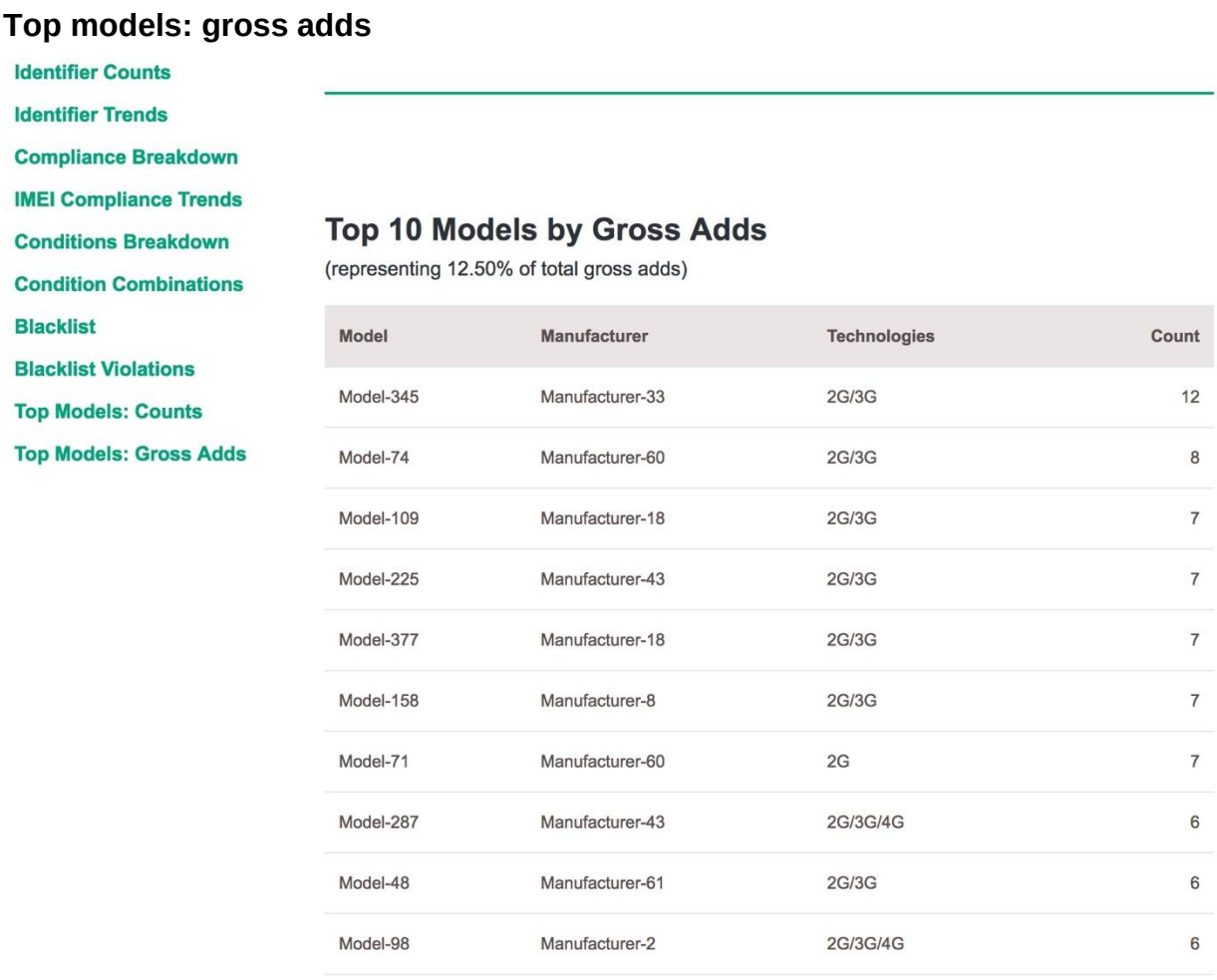


Figure 4-10 Top models: gross adds

4.1.1.2 JSON

The JSON country report covers:

- Blacklist information
- Classification conditions
- Compliance breakdown
- Condition combinations
- Conditions breakdown
- Report name
- Historic blacklist adds
- Historic compliance breakdown
- Historic conditions breakdown
- Historic IMEI, IMSI, MSISDN, and triplet counts

- IMEI/IMSI and IMSI/IMEI overloading
- Daily counts for IMEIs, IMSIs, and MSISDNs
- Top models
- Monthly counts

Blacklist information

Table 4-1 Blacklist information

Field names	Description
blacklist_adds	▪ imeis: Number of IMEIs seen on network during this month which were first blocked during the month ▪ non_paired_imsis: Number of non-paired IMSIs seen on this network during this month associated with those IMEIs
blacklistviolations_by_age	Count of IMEIs seen after they were blacklisted, bucketed by the difference in days between when they were last seen during this month and the block date

```
"blacklist_adds": {
        "imeis": 0,
        "non_paired_imsis": 0
    },
    "blacklist_violations_by_age": {
        "1-2": 0,
        "11-20": 0,
        "21-30": 0,
        "3-5": 0,
        "31-90": 0,
        "6-10": 0,
        "90+": 0
    },
```

Classification conditions

Classification number counts are always filtered by data appearing for that operator or for the whole country during the reporting month. For example, if there are 10 GSMA Not Found IMEIs, but only 6 were seen on an operator network during the month, the report will return 6.

The latest classification data is always used, not taken at the end of the reporting month. If a report was run in September for June, it would first take the IMEIs classified at the current date in September. To generate an IMEI count, it takes the subset of what appeared on the network during that month for that operator.

The following classification conditions were configured at the last dirbs-classify execution.

```
        "classification_conditions": [
        {
            "blocking": true,
            "config": {
                "blocking": true,
                "dimensions": [
                    {
                        "invert": false,
                        "module": "gsma_not_found",
                        "parameters": {}
                    }
```



```
                ],
                "grace_period_days": 90,
                "label": "gsma_not_found",
                "max_allowed_matching_ratio": 0.1,
                "reason": "TAC not found in GSMA TAC database",
                "sticky": false
            },
            "label": "gsma_not_found"
        },
        {
            "blocking": true,
            "config": {
                "blocking": true,
                "dimensions": [
                    {
                        "invert": false,
                        "module": "stolen_list",
                        "parameters": {}
                    }
                ],
                "grace_period_days": 0,
                "label": "local_stolen",
                "max_allowed_matching_ratio": 0.1,
                "reason": "IMEI found on local stolen list",
                "sticky": false
            },
            "label": "local_stolen"
        },
        {
            "blocking": true,
            "config": {
                "blocking": true,
                "dimensions": [
                    {
                        "invert": false,
                        "module": "malformed_imei",
                        "parameters": {}
                    }
                ],
                "grace_period_days": 0,
                "label": "malformed_imei",
                "max_allowed_matching_ratio": 0.1,
                "reason": "Invalid characters detected in IMEI",
                "sticky": false
            },
            "label": "malformed_imei"
        },
        {
            "blocking": false,
            "config": {
                "blocking": false,
                "dimensions": [
                    {
                        "invert": false,
                        "module": "duplicate_threshold",
                        "parameters": {
                            "period_days": 120,
                            "threshold": 3
                        }
                    }
                ],
                "grace_period_days": 90,
                "label": "duplicate_mk1",
                "max_allowed_matching_ratio": 0.1,
```



```
                "reason": "Duplicate threshold exceeeded",
                "sticky": false
            },
            "label": "duplicate_mk1"
        }
```


Compliance breakdown

High level compliance data can be found in the compliance_breakdown field. The breakdown lists stats on overall compliance across all configured conditions.

Table 4-2 lists the properties of the configured conditions. There is also historical data for compliance breakdown in the historic_compliance_breakdown field.

Table 4-2 Compliance breakdown

Field names	Description
num_compliant_imei_imsis	Number of compliant IMEI-IMSIs (no NULLs)
num_compliant_imei_msisdns	Number of compliant IMEI-MSISDNs (no NULLs)
num_compliant_imeis	Number of compliant IMEIs (no NULLs)
num_compliant_triplets	Number of compliant IMEI-IMSI-MSISDN triplets (no NULLs)
num_noncompliant_imei_imsis_blocking	IMEI-IMSIs (no NULLs) meeting 1+ blocking condition
num_noncompliant_imei_imsis_info_only	As above, but meeting only non-blocking conditions
num_noncompliant_imei_imsis	Sum of above 2 counts. Redundant
num_noncompliant_imei_msisdns_blocking	IMEI-MSISDNs (no NULLs) meeting 1+ blocking condition
num_noncompliant_imei_msisdns_info_only	As above, but meeting only non-blocking conditions
num_noncompliant_imei_msisdns	Sum of above 2 counts. Redundant
num_noncompliant_imeis_blocking	IMEIs (no NULLs) meeting 1+ blocking condition
num_noncompliant_imeis_info_only	As above, but meeting only non-blocking conditions
num_noncompliant_imeis	Sum of above 2 counts. Redundant
num_noncompliant_triplets_blocking	Triplets (no NULLs) meeting 1+ blocking condition
num_noncompliant_triplets_info_only	As above, but meeting only non-blocking conditions
num_noncompliant_triplets	Sum of above 2 counts. Redundant


```
    "compliance_breakdown": {
        "num_compliant_imei_imsis": 572,
        "num_compliant_imei_msisdns": 572,
        "num_compliant_imeis": 530,
```


Condition combinations

Table 4-3 lists the stats for every combination of conditions in the condition_combination_table field.

Table 4-3 Condition combinations

Field names	Description
combination	▪ Describes combination of conditions for this entry in the list ▪ If Cond A is True and the rest are False, it means Cond A only ▪ If Cond A and Cond B are true and the rest are False, it means Cond A and Cond B only
compliance_level	▪ 2 means compliant ▪ 1 means non-compliant but informational ▪ 0 means non-compliant and blocking ▪ This is determined by the config of the conditions selected by combination
num_imeis	Number of matching IMEIs (no NULLs)
num_imei_gross_adds	Number of matching gross add IMEIs (no NULLs)
num_imei_imsis	Number of matching IMEI-IMSIs (no NULLs)
num_imei_msisdns	Number of matching IMEI-MSISDNs (no NULLs)
num_subscriber_triplets	Number of matching triplets (no NULLs)


```
    "condition_combination_table": [
        {
            "combination": {
                "duplicate_mk1": false,
                "gsma_not_found": false,
                "local_stolen": false,
                "malformed_imei": false
            },
            "compliance_level": 2,
            "num_imei_gross_adds": 530,
            "num_imei_imsis": 572,
            "num_imei_msisdns": 572,
            "num_imeis": 530,
            "num_subscriber_triplets": 572
        },
        {
            "combination": {
                "duplicate_mk1": true,
                "gsma_not_found": false,
                "local_stolen": false,
                "malformed_imei": false
            },
            "compliance_level": 1,
            "num_imei_gross_adds": 13,
            "num_imei_imsis": 47,
            "num_imei_msisdns": 47,
            "num_imeis": 13,
            "num_subscriber_triplets": 47
        },
```


Conditions breakdown

Overall stats about an individual condition can be found in the conditions_breakdown field. There is an entry for each classification condition (matches label in classification_conditions). There is also historical data for this in the historic_conditions_breakdown field

Table 4-4 Conditions breakdown

Field names	Description
num_imeis	Number of matching IMEIs (no NULLs)
num_imei_gross_adds	Number of matching gross add IMEIs (no NULLs)
num_imei_imsis	Number of matching IMEI-IMSIs (no NULLs)
num_imei_msisdns	Number of matching IMEI-MSISDNs (no NULLs)


```
"conditions_breakdown": {
        "duplicate_mk1": {
            "num_imei_gross_adds": 16,
            "num_imei_imsis": 56,
            "num_imei_msisdns": 56,
            "num_imeis": 16,
            "num_triplets": 56
        },
        "gsma_not_found": {
            "num_imei_gross_adds": 31,
            "num_imei_imsis": 39,
            "num_imei_msisdns": 39,
            "num_imeis": 31,
            "num_triplets": 39
        },
        "local_stolen": {
            "num_imei_gross_adds": 10,
            "num_imei_imsis": 13,
            "num_imei_msisdns": 13,
            "num_imeis": 10,
            "num_triplets": 13
        },
        "malformed_imei": {
            "num_imei_gross_adds": 0,
            "num_imei_imsis": 0,
            "num_imei_msisdns": 0,
            "num_imeis": 0,
            "num_triplets": 0
        }
    },
```


Report name


```
"country_name": "Country1",
    "creation_date": "2017-09-27",
    "end_date": "2017-08-31",
    "has_compliance_data": true,
    "has_data": true,
```


Historic blacklist adds

Historic stats for the last five months of blacklist adds. This is used to generate drawing trends.


```
"historic_blacklist_adds": [
        {
            "imeis": 0,
            "non_paired_imsis": 0
        },
        {
            "imeis": 0,
            "non_paired_imsis": 0
        },
```



```
        {
```

Historic compliance breakdown

```
    "historic_compliance_breakdown": [
{
            "num_compliant_imei_imsis": 572,
            "num_compliant_imei_msisdns": 572,
            "num_compliant_imeis": 530,
            "num_compliant_triplets": 572,
            "num_noncompliant_imei_imsis": 146,
            "num_noncompliant_imei_imsis_blocking": 52,
            "num_noncompliant_imei_imsis_info_only": 47,
            "num_noncompliant_imei_msisdns": 52,
            "num_noncompliant_imei_msisdns_blocking": 52,
            "num_noncompliant_imei_msisdns_info_only": 47,
            "num_noncompliant_imeis": 54,
            "num_noncompliant_imeis_blocking": 41,
            "num_noncompliant_imeis_info_only": 13,
            "num_noncompliant_triplets": 99,
            "num_noncompliant_triplets_blocking": 52,
            "num_noncompliant_triplets_info_only": 47
        }
    ],
```

Historic conditions breakdown

```
    "historic_conditions_breakdown": {
        "duplicate_mk1": [
            {
                "num_imei_gross_adds": 0,
                "num_imei_imsis": 0,
                "num_imei_msisdns": 0,
                "num_imeis": 0,
                "num_triplets": 0
            },
…
            {
                "num_imei_gross_adds": 16,
                "num_imei_imsis": 56,
                "num_imei_msisdns": 56,
                "num_imeis": 16,
                "num_triplets": 56
            }
        ],
        "gsma_not_found": [
            {
                "num_imei_gross_adds": 0,
                "num_imei_imsis": 0,
                "num_imei_msisdns": 0,
                "num_imeis": 0,
                "num_triplets": 0
            },
…
            {
                "num_imei_gross_adds": 31,
                "num_imei_imsis": 39,
                "num_imei_msisdns": 39,
                "num_imeis": 31,
                "num_triplets": 39
            }
        ],
        "local_stolen": [
            {
```



```
                    "num_imei_gross_adds": 0,
                    "num_imei_imsis": 0,
                    "num_imei_msisdns": 0,
                    "num_imeis": 0,
                    "num_triplets": 0
                },
…
                {
                    "num_imei_gross_adds": 10,
                    "num_imei_imsis": 13,
                    "num_imei_msisdns": 13,
                    "num_imeis": 10,
                    "num_triplets": 13
                }
            ],
            "malformed_imei": [
                {
                    "num_imei_gross_adds": 0,
                    "num_imei_imsis": 0,
                    "num_imei_msisdns": 0,
                    "num_imeis": 0,
                    "num_triplets": 0
                },
…
                    "num_imei_gross_adds": 0,
                    "num_imei_imsis": 0,
                    "num_imei_msisdns": 0,
                    "num_imeis": 0,
                    "num_triplets": 0
                }
            ]
        },
```


Historic IMEI, IMSI, MSISDN and triplet counts

Table 4-5 Historic IMEI, IMSI, MSISDN and triplet counts

Field names	Description
historic_imei_counts	Contains list of total_imeis_seen results for previous months for drawing trends
historic_imsi_counts	Contains list of total_imsis_seen results for previous months for drawing trends
historic_msisdn_counts	Contains list of total_msisdns_seen results for previous months for drawing trends
historic_triplet_counts	Contains list of total_triplets_seen results for previous months for drawing trends


```
        "historic_imei_counts": [
            0,
            0,
            0,
            0,
            0,
            584
        ],
        "historic_imsi_counts": [
            0,
            0,
            0,
```



```
            0,
            0,
            671
        ],
        "historic_msisdn_counts": [
            0,
            0,
            0,
            0,
            0,
            671
        ],
        "historic_triplet_counts": [
            0,
            0,
            0,
            0,
            0,
            671
        ],
```


IMEI/IMSI and IMSI/IMEI overloading

Table 4-6 IMEI/IMSI and IMSI/IMEI overloading

Field names	Description
imsi_imei_overloading	Number of IMSIs seen with 1 IMEI, 2 IMEIs, 3 IMEIs, etc.
imei_imsi_overloading	Number of IMEIs seen with 1 IMSI, 2 IMSIs, 3 IMSIs, etc. (duplication)


```
    "imei_imsi_overloading": [
            {
                "num_imeis": 521,
                "seen_with_imsis": 1
            },
            {
                "num_imeis": 47,
                "seen_with_imsis": 2
            },
            {
                "num_imeis": 13,
                "seen_with_imsis": 3
            },
            {
                "num_imeis": 1,
                "seen_with_imsis": 4
            },
            {
                "num_imeis": 1,
                "seen_with_imsis": 5
            },
            {
                "num_imeis": 1,
```



```
            "seen_with_imsis": 8
        }
    ],
```

Daily counts for IMEIs, IMSIs and MSISDNs

Table 4-7 Daily counts for IMEIs, IMSIs and MSISDNs

Field names	Description
imeis_per_day	Distinct IMEIs seen per day (no NULLs)
imsis_per_day	Distinct IMSIs seen per day (no NULLs)
msisdns_per_day	Distinct MSISDNs seen per day (no NULLs)
recs_per_day	Distinct triplets seen per day (no NULLs)

```
    "imeis_per_day": [
        {
            "count": 281,
            "date": "2017-08-01"
        },
…
        {
            "count": 266,
            "date": "2017-08-31"
        }
          "imsi_imei_overloading": [
        {
            "num_imsis": 671,
            "seen_with_imeis": 1
        }
…
    "imsis_per_day": [
        {
            "count": 299,
            "date": "2017-08-01"
        },
…
        {
            "count": 281,
            "date": "2017-08-31"
        },
    "msisdns_per_day": [
        {
            "count": 299,
            "date": "2017-08-01"
        },
…
        {
            "count": 250,
            "date": "2017-08-31"
        },
    "recs_per_day": [
        {
            "count": 299,
            "date": "2017-08-01"
        },
…
        {
            "count": 250,
            "date": "2017-08-31"
        },
```

Report schema version and DIRBS core software version

```
    "report_schema_version": 2,
    "software_version": "5.2.0",
    "start_date": "2017-08-01",
```


Top models

Table 4-8 Top models

Field names	Description
top_models_gross_adds	List of top 10 models by ordered by IMEI gross adds. Each list item contains manufacturer, model, gross add IMEI count and tech generation (2G, etc.)
top_models_imei	List of top 10 models by ordered by raw IMEI count. Each list item contains manufacturer, model, gross add IMEI count and tech generation (2G, etc.)
top_models_gross_adds_count	Sum of IMEI gross add counts for top_models_gross_adds. Used for percentage calculations. Technically redundant
top_models_imei_count	Sum of IMEI gross add counts for top_models_imei. Used for percentage calculations. Technically redundant


```
"top_models_gross_adds": [
        {
            "count": 12,
            "manufacturer": "Manufacturer-33",
            "model": "Model-345",
            "tech_generations": "2G/3G"
        },
        {
            "count": 8,
            "manufacturer": "Manufacturer-60",
            "model": "Model-74",
            "tech_generations": "2G/3G"
        },
        {
            "count": 7,
            "manufacturer": "Manufacturer-18",
            "model": "Model-109",
            "tech_generations": "2G/3G"
        },
…
"top_models_gross_adds_count": 73,
    "top_models_imei": [
        {
            "count": 12,
            "manufacturer": "Manufacturer-33",
            "model": "Model-345",
            "tech_generations": "2G/3G"
        },
        {
            "count": 8,
            "manufacturer": "Manufacturer-60",
            "model": "Model-74",
            "tech_generations": "2G/3G"
        },
        {
            "count": 7,
            "manufacturer": "Manufacturer-18",
            "model": "Model-109",
            "tech_generations": "2G/3G"
        },
…
    "top_models_imei_count": 73,
```


Monthly counts

Table 4-9 Monthly counts

Field names	Description
total_imeis_seen	Number of distinct IMEIs seen (no NULLs)
total_imsis_seen	Number of distinct IMSIs seen (no NULLs)
total_msisdns_seen	Number of distinct MSISDNs seen (no NULLs)
total_imei_imsis_seen	Number of distinct IMEI-IMSI pairs (no NULLs)
total_imei_msisdns_seen	Number of distinct IMEI-MSISDN pairs (no NULLs)
total_imsi_msisdns_seen	Number of distinct IMSI-MSISDN pairs (no NULLs)
total_gross_adds	Number of IMEI gross adds
total_records_seen	Blind COUNT(*) of all rows of data
total_triplets_seen	Number of distinct IMEI-IMSI-MSISDN triplets (no NULLS)
total_null_imei_records	Rows of data containing a NULL IMEI
total_null_imsi_records	Rows of data containing a NULL IMSI
total_null_msisdn_records	Rows of data containing a NULL MSISDN
total_invalid_imei_imsis	Distinct IMEI-IMSI pairs where IMEI or IMSI is NULL
total_invalid_imei_msisdns	Distinct IMEI-IMSI pairs where IMEI or MSISDN is NULL
total_invalid_triplets	Distinct IMEI-IMSI-MSISDN triplets where any is NULL
total_whitespace_imsi_records	Will always be zero in recent release (REMOVE)
total_whitespace_msisdn_records	Will always be zero in recent release (REMOVE)
historic_blacklist_adds	Historic stats for the last five months for above for drawing trends


```
    "total_blacklist_violations": 0,
    "total_gross_adds": 584,
    "total_imei_imsis_seen": 671,
    "total_imei_msisdns_seen": 671,
    "total_imeis_seen": 584,
    "total_imsi_msisdns_seen": 671,
    "total_imsis_seen": 671,
    "total_invalid_imei_imsis": 0,
    "total_invalid_imei_msisdns": 0,
    "total_invalid_triplets": 0,
    "total_msisdns_seen": 671,
    "total_null_imsis": 0,
    "total_null_msisdns": 0,
    "total_records_seen": 671,
    "total_triplets_seen": 671,
    "total_whitespace_imsis": 0,
    "total_whitespace_msisdns": 0
```


4.1.1.3 CSV

Country1_8_2017.csv

Country1_8_2017.csv shows conditions met per TAC and the additional data in the header.

```
TAC,gsma_not_found,local_stolen,malformed_imei,duplicate_mk1,IMEI count,IMEI gross
adds count,IMEI-IMSI count,IMEI-MSISDN count,Subscriber triplet count,Compliance
Level
35929705,False,False,False,False,2,2,2,2,2,2
```



```
35347306,False,False,False,False,1,1,1,1,1,2
35544905,False,False,False,False,1,1,1,1,1,2
35295707,False,False,False,False,1,1,1,1,1,2
35305902,False,False,False,False,1,1,1,1,1,2
35211906,False,False,False,False,1,1,1,1,1,2
35627206,False,False,False,False,1,1,1,1,1,2
35730805,False,False,False,False,1,1,1,1,1,2
…
```

Country1_8_2017_condition_counts.csv

Country1_8_2017_condition_counts.csv shows all configured conditions and additional data in the header.

```
gsma_not_found,local_stolen,malformed_imei,duplicate_mk1,IMEI count,IMEI gross adds
count,IMEI-IMSI count,IMEI-MSISDN count,Subscriber triplet count,Compliance Level
False,False,False,False,530,530,572,572,572,2
False,False,False,True,13,13,47,47,47,1
False,True,False,False,9,9,10,10,10,0
False,True,False,True,1,1,3,3,3,0
True,False,False,False,29,29,33,33,33,0
True,False,False,True,2,2,6,6,6,0
```

4.1.2 Operator reports

HTML and JSON operator reports are identical.

4.1.2.1 CSV

Country1_operator1_8_2017.csv shows conditions met per TAC and additional data in the header.

Country1_operator1_8_2017.csv

```
TAC,gsma_not_found,local_stolen,malformed_imei,duplicate_mk1,IMEI
count,IMEI gross adds count,IMEI-IMSI count,IMEI-MSISDN count,Subscriber
triplet count,Compliance Level
99000435,False,False,False,False,1,1,1,1,1,2
01140800,False,False,False,False,1,1,1,1,1,2
86809701,False,False,False,True,1,1,2,2,2,1
86809701,False,False,False,False,1,1,1,1,1,2
```

Country1_operator1_8_2017_condition_counts.csv

Country1_operator1_8_2017_condition_counts.csv shows all configured conditions and additional data in the header.

```
gsma_not_found,local_stolen,malformed_imei,duplicate_mk1,IMEI count,IMEI
gross adds count,IMEI-IMSI count,IMEI-MSISDN count,Subscriber triplet
count,Compliance Level
False,False,False,False,488,488,488,488,488,2
True,False,False,False,25,25,25,25,25,0
False,True,False,True,2,2,5,5,5,0
False,False,False,True,55,55,131,131,131,1
```

4.2 Condition IMEI overlaps reports

Condition IMEI overlaps reports generate per-condition reports showing matched IMEIs seen on more than one MNO network.

Country1_8_2017_condition_imei_overlap_duplicate_mk1.csv

```
IMEI,Operators
01170100000001,operator1|operator2
01206400000001,operator1|operator2
01219000000001,operator1|operator2
01223745000001,operator1|operator2
```

Country1_8_2017_condition_imei_overlap_gsma_not_found.csv

```
IMEI,Operators
01134900000001,operator1|operator2
01223745000001,operator1|operator2
01349800000001,operator1|operator2
01392300000001,operator1|operator2
```

Country1_8_2017_condition_imei_overlap_local_stolen.csv

```
IMEI,Operators
01368900000001,operator1|operator2
01388500000001,operator1|operator2
01453800000001,operator1|operator2
35236005000001,operator1|operator2
```

Country1_8_2017_condition_imei_overlap_malformed_imei.csv

```
IMEI,Operators
0113AA00000001,operator1|operator2
0122AA45000001,operator1|operator2
0136AA00000001,operator1|operator2
0138AA00000001,operator1|operator2
```

4.3 GSMA not found reports

Country report with list of IMEIs seen on the network that are not found in the GSMA TAC:

Country1_8_2017_gsma_not_found.csv

```
IMEI
01134900000001
01134900000001
01223745000001
01223745000001
```

4.4 Stolen violations reports

Stolen violations reports generate a per-MNO list of IMEIs seen on the network after they were reported stolen.

stolen_violations_operator1.csv

```
imei_norm,last_seen,reporting_date
35236005000001,20170831,20170809
35930705000001,20170831,20170809
35819806000002,20170829,20170809
35570805000002,20170831,20170809
```

stolen_violations_operator2.csv

```
imei_norm,last_seen,reporting_date
35819806000002,20170829,20170809
35793806000001,20170830,20170809
01388500000001,20170829,20170809
01453800000001,20170831,20170809
```

4.5 Top duplicates reports

Country reports of all IMEIs seen with more than five IMSIs:

```
Country1_8_2017_duplicates.csv
IMEI,IMSI count
01206400000001,16
35177105000001,10
35840304000001,8
0138150000000
```

5 Understanding DIRBS Lists

The `dirbs-listgen` command creates .zip files containing both the full lists and all the delta lists in CSV format for blacklists, notifications, and exceptions. ZIP files are named as shown below, where both date_string and operator_id are variables based on the list generation timestamp and the operator id:

- <date string>_blacklist.zip (same for every MNO)
- <date_string>_notifications_<operator_id>.zip
- <date_string>_exceptions_<operator_id>.zip

Full lists contain all the entries that are on the respective list, while the delta list only contains changes between the list-generation runs

E:Running `listgen` with no explicit curr-date parameter bases the end of its lookback window off the most recent operator data date rather than the current date.

5.1 Blacklist

The <date_string> blacklist.zip file will contain the full blacklist and the delta blacklists .CSVs as listed in the sample filenames below. The same blacklists are distributed to all operators,

- 20180217_000302_blacklist.csv
- 20180217_000302_blacklist_delta_-1_42_blocked.csv
- 20180217_000302_blacklist_delta_-1_42_changed.csv

`20180217_000302_blacklist_delta_-1_42_unblocked.csv`

5.1.1 Full blacklist

The full blacklist file will contain the following information:

- Lists IMEIs that have met a blocking condition and where the current date has exceeded the block date.
- A CSV file containing the complete blacklist is distributed to all MNOs.
- One row per IMEI containing these fields:
  - IMEI
  - Block date for IMEI (earliest block date for all the blocking classification conditions that the IMEI meets)
  - List of reasons for this condition (one reason for each condition resulting in the IMEI being blocked, pipe-separated)

20180217_000302_blacklist.csv

```
imei,block_date,reasons
31111106045110,20160503,TAC not found in GSMA TAC database
41111101365980,20160503,TAC not found in GSMA TAC database
12640904324427,20171016,IMEI found on local stolen list
12909602872723,20171016,IMEI found on local stolen list
```

5.1.2 Delta blacklist

Delta blacklists will also be included in the .zip file.

Each file contains the difference between the results of previous list generation run ID for each event type. The file format is:

```
<date_string>_blacklist_delta_-
<Previous_RunID>_<Current_RunID>_<event_type>.csv
```

The following are sample delta blacklist file names:

- 20180217_000302_blacklist_delta_-1_42_blocked.csv
- 20180217_000302_blacklist_delta_-1_42_changed.csv
- 20180217_000302_blacklist_delta_-1_42_unblocked.csv

The delta blacklist file contains the same fields as the full list.

20180217_000302_blacklist_delta_-1_42_blocked.csv

```
imei,block_date,reasons
31111106045110,20160503,TAC not found in GSMA TAC database
41111101365980,20160503,TAC not found in GSMA TAC database
12640904324427,20171016,IMEI found on local stolen list
12909602872723,20171016,IMEI found on local stolen list
```

Table 5-1 Blacklist event types

Event	Example scenarios
blocked	▪ Grace period for an IMEI has expired on a previously-met condition ▪ IMEI meets a blocking condition for the first time and grace period was 0 ▪ IMEI was on the golden list, was meeting a blocking condition and then golden list entry was removed
unblocked	▪ IMEI was previously blocked but no longer meets any blocking condition (non-sticky blocking condition) ▪ IMEI was added to the golden list
changed	▪ IMEI previously was blacklisted, but reasons or block date changed ▪ Perhaps stolen and GSMA Not Found and then the TAC got allocated in GSMA so that the new reasons are just stolen

5.2 Notifications lists

The `<date_string>_notifications_<operator_id>.zip` file will be generated for each operator and contain the full notification and the delta notification CSVs, as listed in the following sample file names:

- 20180217_000302_notifications_operator1.csv
- 20180217_000302_notifications_operator1_delta_-1_42_blacklisted.csv
- 20180217_000302_notifications_operator1_delta_-1_42_changed.csv
- 20180217_000302_notifications_operator1_delta_-1_42_new.csv
- 20180217_000302_notifications_operator1_delta_-1_42_no_longer_seen.csv
- 20180217_000302_notifications_operator1_delta_-1_42_resolved.csv

5.2.1 Full notification list

The full notification list file will contain the following information:

- Lists IMEIs that have met a blocking condition where the current date is still within the grace period for the condition. Does not include any IMEI already on the blacklist.
- For each IMEI, subscriber triplets are generated based on imported operator data. There is one row in the list for each triplet.
- Determines the home network for each triplet based on IMSI and configured MCC/MNC pairs for each configured operator.
- If a triplet does not match any MCC/MNC pairing for a configured operator (roamers, etc.), we notify all operators whose data they have been seen in.
- Each operator gets a different list containing their subscribers and any fallback triplets seen on their network.
- These fields are included in each row:
  - IMEI
  - IMSI
  - MSISDN (if available in country)
  - Block date for IMEI (earliest block date for all the blocking classification conditions that the IMEI meets)
  - List of reasons for this condition (one reason for each condition met by the IMEI, pipe-separated)
  - Amnesty granted field (set to either True or False)
    - Specifies if IMEI is eligible for amnesty

20180217_000302_notifications_operator1.csv

```
imei,imsi,msisdn,block_date,reasons, amnesty_granted
38674133009747,11101536296900,22300001929746,20161206,IMEI not found on
local registration list,blacklisted,false
```

5.2.2 Delta notification lists

Delta notification lists will also be included in the .zip file.

Each file will contain the difference between the results of previous list generation run ID for each event type. The file format is:

```
<date_string>_notification_<operator_id>_delta_-
<Previous_RunID>_<Current_RunID>_<event_type>.csv
```

The following are sample delta notification list file names:

- 20180217_000302_notifications_operator1_delta_-1_42_blacklisted.csv
- 20180217_000302_notifications_operator1_delta_-1_42_changed.csv
- 20180217_000302_notifications_operator1_delta_-1_42_new.csv
- 20180217_000302_notifications_operator1_delta_-1_42_no_longer_seen.csv
- 20180217_000302_notifications_operator1_delta_-1_42_resolved.csv

The delta notification list will contain the same fields as the full list.

20171208_235247_notifications_operator1_delta_36_41.csv

```
imei,imsi,msisdn,block_date,reasons,amnesty_granted
38674133009747,11101536296900,22300001929746,20161206,IMEI not found on
local registration list,blacklisted,false
```

Table 5-2 Notification list event types

Event	Example scenarios
new	▪ IMEI has met a blocking condition for the first time, and there is a non-zero grace period ▪ A new subscriber triplet has been seen with an IMEI meeting a blocking condition and in grace period (changed SIM) ▪ Pairing has been removed for a subscriber using an IMEI meeting a blocking condition and in grace period ▪ IMSI did not have an identifiable home network (via MCC-MNC) and gfipldg was seen for the first time on a network in the lookback window
resolved	▪ IMEI no longer meets a blocking condition and was in grace period previously ▪ Pairing added for a subscriber using an IMEI meeting a blocking condition and in grace period ▪ IMEI added to golden list and was in grace period previously ▪ Triplet no longer seen during lookback window so no longer needs to be notified
blacklisted	▪ IMEI met a blocking condition and was in grace period, but now grace period has expired ▪ IMEI met a new blocking condition that had 0 grace period
changed	IMEI is in grace period, but reasons or block date changed (blocking condition added or removed since last list generation)
no_longer_seen	The triplet was removed from the notifications list, is not paired or blacklisted but the IMEI is still being notified

5.3 Exceptions lists

The `<date_string>_exceptions_<operator_id>.zip` file will be generated for each operator and contain the full exception and the delta exceptions CSVs, as listed in the following sample filenames:

- 20180217_000302_exceptions_operator1.csv
- 20180217_000302_exceptions_operator1_delta_-1_42_added.csv
- 20180217_000302_exceptions_operator1_delta_-1_42_removed.csv

5.3.1 Full exceptions list

A full exceptions list file contains the following information:

- Each operator gets a copy of the pairing list, split into per-operator exception lists based again on their IMSI and the configured MCC/MNC pairs for the configured operators.
- If a pairing's IMSI matches any MCC/MNC pairing for a configured operator (roamers, etc.), the pairing is placed on each operator's exception lists which that IMEI/IMSI combination has been seen.
- These fields are included in each row:
  - IMEI
  - IMSI

20180217_000302_exceptions_operator1.csv

```
imei,imsi
811111013136464,111038001111111
311111060451100,111035111111111
411111013659808,310035111111111
```

5.3.2 Delta exceptions list

Delta exceptions lists will also be included in the .zip file.

Each file contains the difference between the results of previous list generation run ID for each event type. The file format is:

```
<date_string>_exceptions_<operator_id>_delta_-
<Previous_RunID>_<Current_RunID>_<event_type>.csv
```

The following are sample delta notification list file names:

- 20180217_000302_exceptions_operator1_delta_-1_42_added.csv
- 20180217_000302_exceptions_operator1_delta_-1_42_removed.csv

20180217_000302_exceptions_operator1_delta_1_42.csv

```
imei,imsi,change_type
64220299727231,111041012987198,added
```

Table 5-3 Exceptions list change types

Event	Example scenarios
added	▪ IMEI-IMSI pair has been added to the pairing list since last run ▪ IMSI did not have an identifiable home network (via MCC-MNC) and was seen for the first time on a network ▪ Config setting restrict_exceptions_list_to_blacklisted_imeis is True and the IMEI associated with this pairing just got blacklisted
removed	▪ IMEI-IMSI pair has been removed from the pairing list since last run ▪ IMSI did not have an identifiable home network (via MCC-MNC) and was no longer seen on a network ▪ Config setting restrict_exceptions_list_to_blacklisted_imeis is True and the IMEI associated with this pairing was unblocked

6 Understanding DIRBS Whitelist/Restrictive Mode

As of Core v13.0.0, in addition to blacklist/permissive mode, DIRBS allows whitelist/restrictive mode as well.

This gives flexibility to those who want to implement DIRBS as a whitelisting system instead of blacklisting system. Required configuration is described in the config section of this document. For its installation, please refer to the installation guide of DIRBS Core v16.0.0.

In whitelist/restrictive mode, DIRBS allows only whitelisted IMEIs to be operational on the network. Regulator manages white-listed IMEIs centrally. System can be configured in Synchronize mode which distributes the list to all connected MNOs so that they can perform all IMEI checks locally OR Authorize mode to allow MNOs to perform IMEI checks from DIRBS central database in the real-time.

The DIRBS Core mainly provides following functionalities for Whitelisting:

- Whitelist processing
- Whitelist distribution

The diagram below depicts flow of the whitelisting process and its functionality:

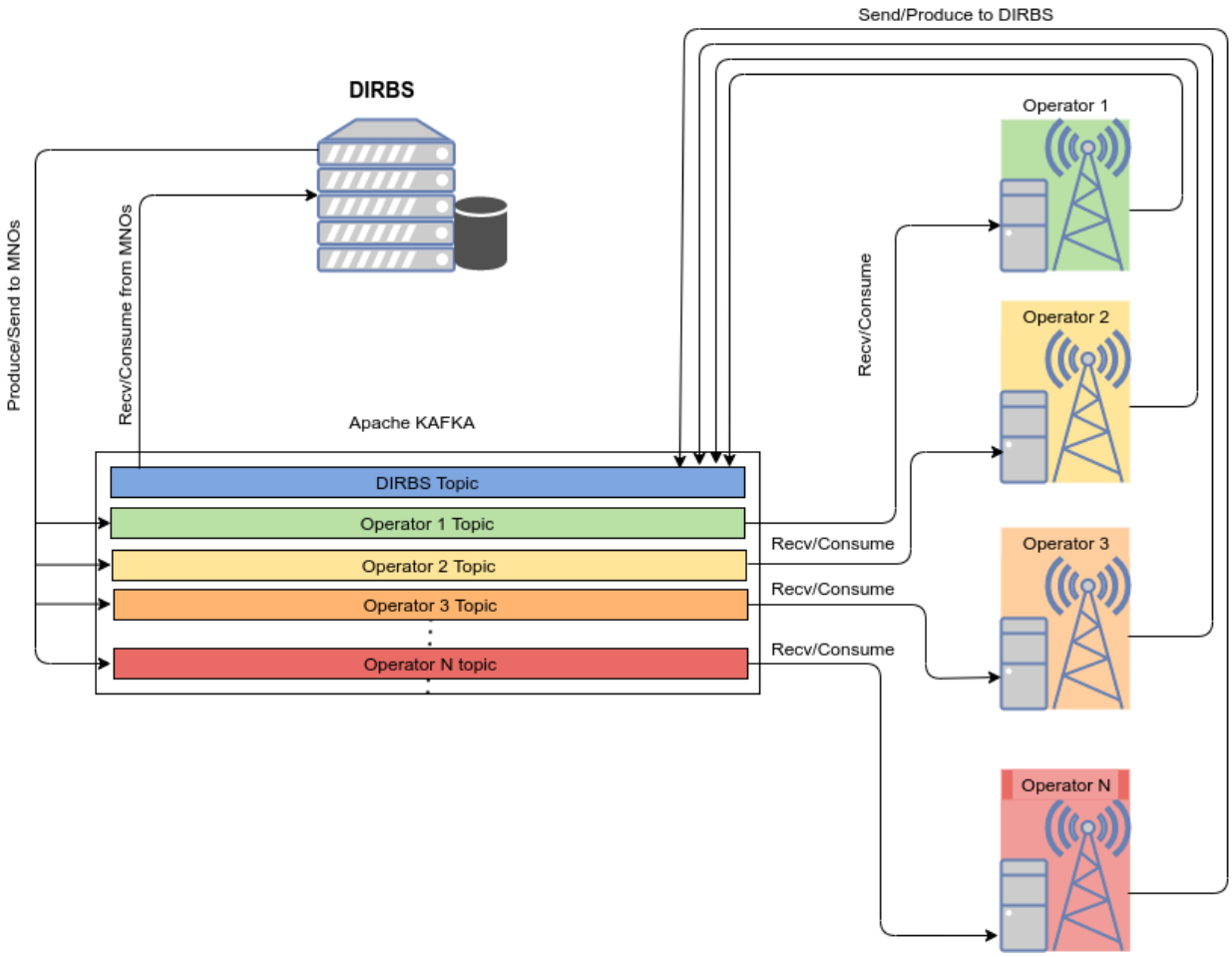


Figure 6-1 DIRBS Whitelist diagram

DIRBS System and MNOs/Operators are connected through a real time data streaming interface i.e. KAFKA, which is to send and receive data in real time. Following is the basic flow of the system.

For example, an **IMEI 123456789012345** is in whitelist but not paired/associated with any IMSI (MSISDN/subscription), the user of that IMEI gets registered with **Operator 1.** The operator 1 is supposed to pair/associate that IMEI by sending a request to DIRBS via **DIRBS Topic**. After receiving the request, DIRBS will validate that request and pair/associate that IMEI with operator 1 and will notify other operators about the IMEI status change via their respective topics.

7 Frequently Asked Questions

7.1 How does duplicate averaging work?

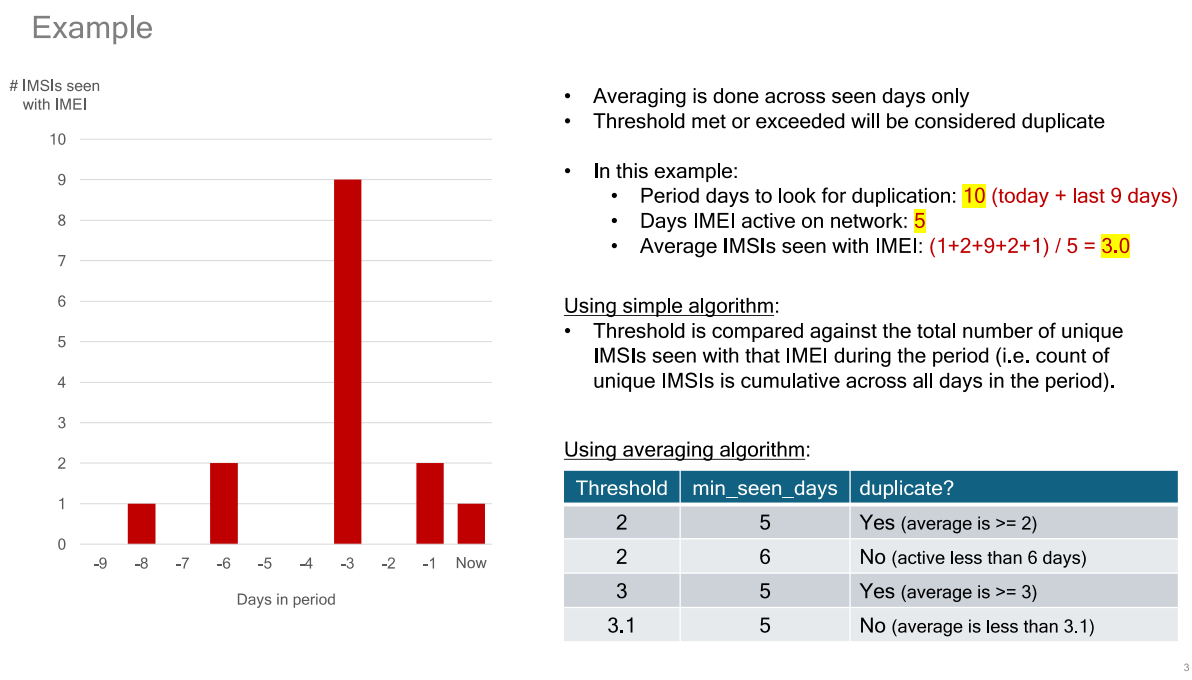


Figure 7-1 Duplicate averaging

7.2 Reported error during dirbs-classify or dirbs-listgen

The following error message is the result of abrupt changes (sanity checks failure) in the core configs which affects classification and list generation:

```
dirbs.listgen.generator.ListGenerationSanityChecksFailedException: Sanity
checks failed, configurations are not identical to the last successful list
generation
```

```
dirbs.cli.classify.ClassifySanityChecksFailedException: Sanity checks
failed, configurations are not identical to the last successful
classification
```

Changes to the following fields can cause this error:

- Classification: operator's config, conditions, amnesty configs
- List Generation: operator's config, lookback days, blocking conditions, amnesty configs

7.3 Reported error during dirbs-classify or dirbs-import

The following error message is the result of a connection timeout between the DIRBS Core and the PostgreSQL server:

```
2017-11-24 13:03:49,826 - dirbs.exception - ERROR - DIRBS encountered an
uncaught software exception
...
psycopg2.DatabaseError: SSL SYSCALL error: Connection timed out
...
psycopg2.OperationalError: SSL SYSCALL error: EOF detected
...
During handling of the above exception, another exception occurred:
...
psycopg2.DatabaseError: SSL SYSCALL error: Connection timed out
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

The timeout can be caused and fixed by either or both of the following:

- PostgreSQL server requires tuning. Logs from the server must be analyzed during the tuning process. Check the following on the PostgreSQL server:
  - tcp_keepalives_count
  - tcp_keepalives_idle
  - tcp_keepalives_interval
- Network device configuration, i.e., firewalls:
  - Increase TCP timeout to greater than 1800

7.4 Reported error during `dirbs-import`

The following error message occurred while importing operator data and is the result of insufficient disk space on the PostgreSQL server:

```
File "/usr/lib/python3.5/concurrent/futures/_base.py", line 357, in
__get_result
    raise self._exception
File "/usr/lib/python3.5/concurrent/futures/thread.py", line 55, in run
    result = self.fn(*self.args, **self.kwargs)
File "/home/dirbs/dirbs-venv/lib/python3.5/site-
packages/dirbs/importer/abstract_importer.py", line 338, in
_upload_file_to_staging_table
    cursor.copy_expert(sql=self._upload_batch_to_staging_table_query(),
file=f)
```



```
psycopg2.OperationalError: could not extend file "base/24702/25222.8":
wrote only 4096 of 8192 bytes at block 1162731
HINT: Check free disk space.
CONTEXT:  COPY staging_operator_import_5, line 320617
```

This issue can be resolved by adding additional disk space to your PostgreSQL Server.

7.5 Understanding gsma_not_found Reporting Body Index delay configuration

The dirbs.yml file enables the configuration of the Reporting Body Index (RBI) delays to be used when classifying the gsma_not_found condition. For syntax and default values, see Appendix B.

Due to delays by the reporting body, there can be a lag between the TAC allocation date and the GSMA TAC DB. New IMEIs may be seen on the network before the TAC is included in the GSMA TAC DB and can be erroneously reported as gsma_not found and potentially prematurely blocked.

The RBI delay enables the configuration of a delay in days on a per RBI basis (see Figure 7-2 RBI delay

). An IMEI that contains an RBI listed in Appendix B will not be classified as gsma_not_found until the RBI delay period has elapsed.

Figure 7-2 RBI delay

The default values configured in the DIRBS Core have been selected based on the analysis of historical data.

E:For all other RBIs that are not listed in Appendix B or configured in the dirbs.yml, the RBI delay is 0. Any found IMEI whose tag does not include a legitimate RBI will be immediately classified as gsma_not_found.

7.6 Duplicate and conflicting rows in non-operator imports

This section provides the rationale behind the "conflicting rows" check that has been implemented in DIRBS 7.0.0 and for which there is no option to disable. This includes:

- The difference between duplicate and conflicting rows, and

- Why DIRBS Core cannot safely import files containing conflicting data for the same records.

7.6.1 Key and metadata columns

Every non-operator import in DIRBS Core (stolen list, pairing list, registration list, and golden list) has columns that fall into two categories:

- **Key columns** uniquely identify the device or pairing. This is often the normalized IMEI for many imports.
- **Metadata columns** contain metadata associated with the device or pairing identified by the key columns. This might be make, model, status, reporting date, etc.

Table 7-1 summarizes the columns for each type of non-operator import.

Table 7-1 Key and metadata columns

Import type	Key columns	Metadata columns in 11.0.0
Stolen list	Normalized IMEI	Reporting date, status
Pairing list	Normalized IMEI, IMSI	None
Registration list	Normalized IMEI	Make, model, status, model_number, brand_name, device_type, radio_interface, device_id
Golden list	Normalized IMEI or Hashed Normalized IMEI	None
Barred list	Normalized IMEI	None
Barred TAC list	TAC	None
Subscribers List	UID, IMSI	None

7.6.1.1 Normalized IMEI – imei_norm

Unfortunately, there is no single definition of an IMEI. There are at least four variants of the same standards-compliant IMEI:

- 14-digit IMEI (no check digit or software version)
- 15-digit IMEI (with Luhn check digit calculated and appended)
- 15-digit IMEI (with 0 transmitted as last digit, as sent over the air)
- 16-digit IMEI (with 2-digit software version appended to 14-digit number)

DIRBS must normalize IMEIs using some well-defined process to ensure that variations listed above map to the same IMEI during classification and list generation.

The algorithm used by DIRBS Core is:

- Trim leading and trailing whitespace to produce TRIMMED_IMEI. If TRIMMED_IMEI is an empty string, convert it to NULL.
  - If TRIMMED_IMEI starts with 14-digit characters (0-9), use those 14 digits as the normalized IMEI.

- Else, return the uppercase version of TRIMMED_IMEI as the normalized IMEI.

7.6.2 Problems

7.6.2.1 Duplicate keys in the file

After normalization of IMEIs, there might be duplicate keys in an input file.

Table 7-2 and Table 7-3 show an example file of a stolen list before and after normalization, respectively.

Table 7-2 Stolen list

imei	reporting_date
123456789012345	2017-01-01
1234567890123463	2017-01-01

Table 7-3 Stolen list after normalization

imei_norm	reporting_date
12345678901234	2017-01-01
12345678901234	2017-01-01

The imei_norm column in Table 7-3 is now duplicated in the file. There may have also been IMEIs in the original file that were duplicated even before normalization.

In general, there are two possible scenarios:

- There are duplicate keys, but all the metadata columns agree, i.e., they all contain the same value. This is the case in Table 7-2 and Table 7-3, where the reporting_date column has the same value for both rows with the duplicate imei_norm. This is called a duplicate row, because it is an exact duplicate of the other row. **These kinds of duplicates can be safely ignored by the importer.**
- There are duplicate keys, but the metadata columns do not agree, i.e., they contain different values. These are called conflicting rows because the rows contain conflicting data for the same key. **These kinds of rows cannot be safely imported and will fail the conflicting rows validation check.**

7.6.2.2 Prior resolution of conflicting rows required

Table 7-4 provides an example of two conflicting rows. These rows cannot be safely imported because DIRBS Core does not know how to resolve the conflict.

Table 7-4 Conflicting rows

imei_norm	reporting_date
12345678901234	2017-01-01
12345678901234	2018-03-02

If the only metadata column is reporting_date, it might be possible to simply take the minimum of the reporting_date and store that, but this may be an incorrect assumption and does not also hold for metadata that are not dates.

Table 7-5 is a hypothetical example of conflicting rows, containing extra columns proposed in a future DIRBS Core release for the registration list.

Table 7-5 Future DIRBS Core registration list

imei_norm	make	model	status
12345678901234	Samsung	Galaxy	whitelist
12345678901234	Apple	iPhone	pending_approval

If the registration_list importer was to import the above rows, how would it know which row to import? DIRBS Core does not have enough information to make a decision.

Why not import both rows?

Importing both rows causes the following issues:

- The key columns (imei_norm) are currently used by the database table as a primary key, meaning that they must be unique. This is only a technical limitation only. The primary key can be changed at the expense of performance and optimal query plans if there is a very good reason to do so.
- If the same IMEI is on the list twice, what make/model should be returned? Is the status of that IMEI that it is registered and therefore whitelisted, or that it is still pending approval from the regulator? Systems like the DVS do not know what answer to return for an IMEI if there is conflicting information for the same IMEI.
- Importing both rows can mess up reporting. If we choose the wrong reporting date, the blacklist violations report might say that there was a stolen list violation that is a false positive.

Delta imports

Delta imports pose an additional, related problem if the same IMEI is in the delta file after normalization with different change types.

For example, after normalizing an IMEI might be 'added' and 'removed' in the same delta file (see Table 7-6).

Table 7-6 Normalized delta file

imei	reporting_date	change_type
12345678901234	2017-01-01	add
123456789012345	2017-01-01	remove

This is another conflict that DIRBS Core cannot resolve. This special check for delta files is called the 'multiple_changes_check' and only performs if the importer is in delta mode. If a delta file import is failing this check, it is a simlar issue to the conflicting rows check above.

7.6.3 Options for resolving a conflicting row problem

There are three options for resolving a conflicting row problem:

- Ensure that subsystems that record IMEIs use the same normalization rules as DIRBS Core.
  - This way there are never any duplicates detected by DIRBS Core that could potentially have conflicting data. If the same IMEI is reported stolen twice, for example, the IMEI should already be blocked and a second report should not be lodged. This would be required for new subsystems using replication to directly replicate into DIRBS Core database tables.
- Do not import with any metadata columns.
  - The subsystems in some deployments may not use extra metadata columns. The only use of metadata columns is in the stolen list import, which uses this information for the stolen_violations report. This might not be a viable option.
- Pre-process data during export for DIRBS.
  - If there is a business need to store duplicate data in the subsystems, these can simply be filtered out during export of data for DIRBS Core using a conflict resolution process based on business rules.

7.7 DIRBS Amnesty feature

The DIRBS Amnesty (grandfathering) feature enables DIRBS operators to grant a grace period for IMEIs that have been seen on the network that match certain blocking and amnesty eligible conditions.

When an IMEI is eligible for amnesty, it will not be blacklisted or blocked from the network for the duration of the amnesty period and enables IMEIs to continue working on the operator network. Once the amnesty period expires, IMEIs are classified based on the rules and conditions configured in .dirbs.yml.

The DIRBS system maintains the list of amnesty eligible IMEIs internally in the DIRBS database.

DIRBS Amnesty contains three phases:

- **Amnesty evaluation:** A pre-configured period of time that determines IMEI amnesty eligibility based on conditions configured in the .yml file. Amnesty eligible IMEIs are determined by running the `dirbs-classify` command.
- **Amnesty:** A pre-configured period of time that defines when amnesty is in effect. Amnesty eligible IMEIs will be on the notification list with a block date that is the amnesty end date (see Section 7.7.0). IMEIs are no longer evaluated for amnesty eligibility.
- **Post-amnesty:** Amnesty period has expired. The system no longer checks whether an IMEI is eligible for amnesty. Normal system classification and notifications are in effect.

7.7.1 Enabling and configuring amnesty in .dirbs.yml

To enable the amnesty feature and configure amnesty evaluation and period end dates:

```
# Definition of settings to be used for amnesty feature. Amnesty feature
enables native grandfathering support within
# DIRBS Core. A list of whitelisted IMEIs is managed within Core
transparent to EIRs during the amnesty period.
# The amnesty list is mutable during the amnesty evaluation period,
immutable during the amnesty period.
# During the amnesty evaluation period, the amnesty_list table is
overwritten each time dirbs-classify is run.
amnesty:
  # Boolean value to indicate whether to enable this feature or not.
  amnesty_enabled: True
  # End date of the amnesty evaluation period & start of the amnesty
period.
  evaluation_period_end_date: 20180131
  # End of amnesty period.  Must be greater than the evaluation period end
date.
  amnesty_period_end_date: 20180417
```

Conditions that determine amnesty eligibility are configured in .dirbs.yml:

```
- label: simple_dimension
    dimensions:
      - module: gsma_not_found
    grace_period_days: 0
    blocking: True
    amnesty_eligible: True
    reason: Violated simple dimension
```

E:The blocking parameter must be set to True for the amnesty eligible parameter to take effect.

7.7.2 Eligibility and notifications

During the amnesty evaluation period, IMEIs will be classified using the configured amnesty eligible conditions. When running `dirbs-classify` and the IMEI meets the amnesty eligible condition, it will be classified as amnesty eligible and stored as such in the DIRBS classification state table.

Running dirbs-listgen during this period will not generate notifications for amnesty eligible IMEIs.

Running dibs-listgen during the amnesty_period will generate notifications for amnesty eligible IMEIs in the format described in Section 5.2.

7.7.2.1 Modifications

Evaluation period

The evaluation period can be extended or reduced by modifying the evaluation_period_end_date and re-running `dirbs-classify`.

Amnesty period

The amnesty_period can be extended or reduced by modifying the amnesty_period_end_date and re-running `dirbs-classify`.

7.7.2.2 Disabling

Amnesty can be disabled prior to the amnesty evaluation end date by toggling the amnesty_enabled parameter in the .dirbs.yml to False.

Changing this value to False after the evaluation period or during the amnesty period does not disable the feature as amnesty eligible IMEIs have already been classified and stored in the database.

During the amnesty period, the feature can be disabled by accelerating the amnesty period end date to the current date and re-running `dirbs-classify`. IMEIs will then be classified based on the configured conditions and notified/blocked accordingly.

7.7.3 Stolen, paired, and golden IMEI interaction

Amnesty eligible IMEIs added to the golden list will not be added to the notification list.

Amnesty eligible IMEIs added to the pairing list will exclude a subset of triplets from the notification list.

Example

Triplets seen on the network:

- IMEI-1 - IMSI-1 - MSISDN-1
- IMEI-1 - IMSI-2 - MSISDN-2
- IMEI-1 - IMSI-3 - MSISDN-3

If IMEI-1 is amnesty eligible and also on the golden list, then none of the above triplets are on the notification list.

If IMEI-1 is amnesty eligible and IMEI-1-IMSI-1 is on the pairing list, then IMEI-1-IMSI-2-MSISDN-2 and IMEI-1-IMSI-3-MSISDN-3 triplets will be on the notification list.

The behavior for amnesty eligible IMEIs that are added to the stolen list must be specifically configured in the conditions section of the .dirbs.yml.

It is recommended to treat stolen (not duplicated) IMEIs, and stolen and duplicated IMEIs differently:

- For an amnesty eligible IMEI that was stolen and not duplicated, the IMEI should be blocked and blacklisted. This should be configured as a compound dimension in the conditions section.

```
- label: compound_dimension1
  dimensions:
    - module: stolen_list
    - module: duplicate_threshold
      parameters:
        threshold: 2
        period_days: 30
      invert: True
  grace_period_days: 0
  blocking: True
  reason: Violated compound dimension stolen & not duplicate
  max_allowed_matching_ratio: 0.1
```

- For an amnesty eligible IMEI that was stolen and duplicated, it is recommended to not block the IMEI so as not to impact other IMEIs that are amnesty eligible.

  It is recommended that the Operator terminate the subscription associated with the stolen device.

```
- label: compound_dimension2
  dimensions:
    - module: stolen_list
    - module: duplicate_threshold
      parameters:
        threshold: 2
        period_days: 30
      invert: False
  grace_period_days: 0
  blocking: False
  amnesty_eligible: Trus
  reason: Violated compound dimension of stolen & duplicate
  max_allowed_matching_ratio: 0.1
```

7.7.3.1 Post-amnesty

After the amnesty period, DIRBS Core will classify IMEIs based on conditions configured in .dirbs.yml. It is recommended that the conditions are reviewed and reconfigured as required.

DIRBS Configuration File Sample: YML

A.1 Sample annotated config for DIRBS Core configuration

```
# (C) 2016-2017 Qualcomm Technologies, Inc.  All rights reserved.
#
# PostgreSQL settings used to build connection string
postgresql:
  # Database name (an empty database on the first run). Overridden by
  # environment
  # variable DIRBS_DB_DATABASE if set.
  database: dirbs
  # Host that the PostgreSQL server runs on. Overridden by environment
  # variable DIRBS_DB_HOST if set.
  host: localhost
  # PostgreSQL port if not running on standard port of 5432. Overridden by
  # environment
  # variable DIRBS_DB_PORT if set.
  port: 5432
  # Database role/user that DIRBS will connect to PostgreSQL as. Overridden
  # by environment
  # variable DIRBS_DB_USER if set.
  user: dirbs
  # Password used to connect to the database.
  #
  # There are a number of ways to set the password, with each option having
  # pros and cons
  # dependent on the level of security required vs. ability to automate
  #   - Firstly, the password can be defined here in clear text. This file
  # would then have its permissions set appropriately to restrict access to
  # non-admin users
  #   - If the setting is not defined in this config file, the
  #     user's .pgpass file will be read from their home directory. Note
  #     that this file will only be read if its permissions are set
  #     appropriately (must only be readable by the user)
  #   - If the DIRBS_DB_PASSWORD environment variable is set, this value
  #     will overwrite any value configured in here or in .pgpass
  #   - Finally, the --db-password-prompt command-line option can be used
  #     to prompt the user for a password when a command is run.
  #
  # Uncomment the below line to set the password explicitly in this config
```



```
  # file
  #
  # password: <change me>

# Definitions of regional settings used by DIRBS core for reporting and
# for input validation.
region:
  # Name is used for the country level report
  name: Country1
  # Whether or not MSISDN data is present and should be imported for this
  # region
  import_msisdn_data: True
  # Whether or not RAT data is present and should be imported for this
  # region
  import_rat_data: True
  # country_codes are used to validate MSISDNs during operator data import
  country_codes:
    - "22"
  # exempted_device_types contains a list of GSMA device types that do not
  # require registration in this country. Specifiying a list of device
  # types here will mean that the not_in_registration_list classification
  # dimension will ignore IMEIs whose TACs correspond to the listed device
  # types. They will also be ignored in the IMEI API's realtime
  # registration check. The expected syntax for this is:
  #
  # exempted_device_types:
  #   - Module
  #   - Tablet
  exempted_device_types: []
  # operators map operator IDs to a more human-friendly display string for
  # reporting purposes
  operators:
    - id: operator1
      name: First Operator
      # mcc_mnc values are used to:
      # - validate IMSIs during operator data import
      # - work out which operators notifications about an offending
      #   subscriber
      #   should be sent to
      # - work out which operators excepted IMEI-IMSI pairings should be
      #   sent to
      mcc_mnc_pairs:
        - mcc: "111"
          mnc: "01"
    - id: operator2
      name: Second Operator
      mcc_mnc_pairs:
        - mcc: "111"
```



```
            mnc: "02"
        - id: operator3
          name: Third  Operator
          mcc_mnc_pairs:
            - mcc: "111"
              mnc: "03"
        - id: operator4
          name: Fourth Operator
          mcc_mnc_pairs:
            - mcc: "111"
              mnc: "04"

    # Definitions of configuration variables related to pruning of subscriber
    # data after a specified retention window
    data_retention:
      # The number of months from the start of the current months that DIRBS
      # core will retain data about a triplet seen in its DB. After this time,
      # the triplet will be erased from the seen_triplet table. The IMEI will
      # continue to be stored after this date as it is needed for continued
      # list generation, etc.
      # All references to IMSI and MSISDN will be pruned after this date.
      months_retention: 6
      # The number of days for which an IMEI in blacklist have not bben active
      # on the network will be expired from the blacklist. The IMEI will be
      # allowed to function on the network again. If it meets a condition again
      # then it will be blocked again.
      #
      # uncomment the below line to set the retention period for blacklisted
      # blacklist_retention: <enable me>

    # Definitions of configuration variables used by DIRBS Core in the list
    # generation process.
    list_generation:
      # The number of days that DIRBS core will look back through data from
      # current date to determine IMSIs/MSISDNs
      # which were associated with the notifiable IMEIs.
      lookback_days: 180
      # If true, the exception list will contain only those IMEI-IMSI pairs
      # where the IMEI is on the blacklist. By default, all IMEI-IMSI pairs
      # part of the pairing list are output to the exception list.
      restrict_exceptions_list_to_blacklisted_imeis: false
      # If true, generate a check digit for IMEIs during list generation.
      # Check digit will only be added to "valid IMEIs"
      generate_check_digit: false
      # If true, output only "valid" IMEIs.
      # Valid IMEIs start with 14 digits as they will have 15 digits if the
      # check digit append has been enabled
      output_invalid_imeis: true
```



```
  # If enabled barred IMEIs will also be included in exceptions lists
  include_barred_imeis_in_exceptions_list: false

# Definitions of configuration variables used by DIRBS Core in the report
# generation process.
report_generation:
  # This setting is used by blacklist violations and stolen list violations
  # reports to give the MNO some processing time (in days) before an IMEI
  # appearing on the network is considered a violation.
  blacklist_violations_grace_period_days: 2

# Definitions of configuration variables used by DIRBS Core to determine
# how many workers to use to parallelise
multiprocessing:
  # The maximum number of local processing blade workers to use to achieve
  # DIRBS Core tasks. This is particularly useful for pre-validation of
  # large operator import jobs where we can run multiple instances of the
  # pre-validator in parallel on different parts of the file. The default
  # is to use half of the available CPUs in the system will be used.
  # max_local_cpus: 10
  # The maximum number of database connections to use to parallelise DIRBS
  # Core tasks. PostgreSQL 9.6 has support for parellelising tasks
  # internally - this setting does not affect parellelisation for a single
  # connection. Where PostgreSQL is unable to parallelise a single query by
  # itself, we use this number of workers to issue multiple queries at once
  # on different connections. Generally, this scales very well - it is safe
  # to set this reasonably high. It should probably be set to roughly the
  # number of disks in your RAID array in case there are I/O intensive DB
  # operations going on. If using SSD, can be set to a higher value.
  max_db_connections: 4

# Definition of ratio limits for the various checks on operator data.
operator_threshold:
  # The proportion of the entries in the data that are allowed to have a
  # NULL IMEI
  null_imei_threshold: 0.05
  # The proportion of the entries in the data that are allowed to have a
  # NULL IMSI
  null_imsi_threshold: 0.05
  # The proportion of the entries in the data that are allowed to have a
  # NULL MSISDN (ignored if MSISDN disabled)
  null_msisdn_threshold: 0.05
  # The proportion of the entries in the data that are allowed to have a
  # NULL RAT (ignored if RAT disabled)
  null_rat_threshold: 0.05
  # The proportion of the entries in the data that are allowed to have any
  # column equal to NULL
  # This only includes columns enabled in the import (MSISDN and RAT may be
```



```
    # excluded)
    null_threshold: 0.05
    # The proportion of the non-NULL IMEIs in the data that are allowed to
    # not start with 14 digits
    unclean_imei_threshold: 0.05
    # The proportion of the non-NULL IMSIs in the data that are allowed to
    # not be 14-15 digits
    unclean_imsi_threshold: 0.05
    # The proportion of entries in the data that are allowed to have either a
    # unclean IMEI or an unclean IMSI
    unclean_threshold: 0.05
    # The proportion of the non-NULL IMSIs in the data that are allowed to
    # have a MCC that does not match the
    # configured region
    out_of_region_imsi_threshold: 0.1
    # The proportion of the non-NULL MSISDNs in the data that are allowed to
    # have a CC that does not match the
    # configured region. Ignored ir MSISDN disabled
    out_of_region_msisdn_threshold: 0.1
    # The combined proportion of entries in the data that are allowed to have
    # either a CC (IMSI) or MCC (MSISDN)
    # that does not match the configured region. Ignored if MSISDN if
    # disabled, as this would then be the same as the out of region IMSI
    # check.
    out_of_region_threshold: 0.1
    # The proportion of the entries in the data that are allowed to have an
    # IMSI not starting with one of the MCC-MNC
    # prefixes associated with the operator the data is being imported for
    non_home_network_threshold: 0.2
    # The minimum valid ratio of average daily IMEI count against historical
    # daily IMEI count for a data dump to be considered valid.
    historic_imei_threshold: 0.9
    # The minimum valid ratio of average daily IMSI count against historical
    # daily IMSI count for a data dump to be considered valid.
    historic_imsi_threshold: 0.9
    # The minimum valid ratio of average daily MSISDN count against
    # historical daily MSISDN count for a data dump to be considered valid.
    # Ignored if MSISDN if disabled
    historic_msisdn_threshold: 0.9

# Each of the following importers specifies 2 historic_thresholds which can
# be used to validate new import row count against previously imported data
# for the same importer.
# - import_size_variation_absolute: The most an import can decrease in
#   absolute row count before it is rejected as invalid. By setting this
#   variable to -1, this check will be disabled.
# - import_size_variation_percent: The most an import can decrease in
#   percentage row count before it is rejected as invalid. 0.75 indicates a
```



```
#   new import must be at least 75% of the previous import's row count or
#   it will be rejected. Therefore, setting this variables to 0 will
#   disable this check.
gsma_threshold:
  import_size_variation_absolute: 100
  import_size_variation_percent: 0

pairing_list_threshold:
  import_size_variation_absolute: 1000
  import_size_variation_percent: 0.95

stolen_list_threshold:
  import_size_variation_absolute: -1
  import_size_variation_percent: 0.75

registration_list_threshold:
  import_size_variation_absolute: -1
  import_size_variation_percent: 0.75

golden_list_threshold:
  import_size_variation_absolute: -1
  import_size_variation_percent: 0.75

barred_list_threshold:
  import_size_variation_absolute: -1
  import_size_variation_percent: 0.75

barred_tac_list_threshold:
  import_size_variation_absolute: -1
  import_size_variation_percent: 0.75

subscribers_list_threshold:
  import_size_variation_absolute: 1000
  import_size_variation_percent: 0.95

association_list_threshold:
  import_size_variation_absolute: 1000
  import_size_variation_percent: 0.95

# Definition of conditions used by the DIRBS system. There are zero or more
# conditions used to drive the classification. A system with zero
# conditions does no classification at all
conditions:
  # Each condition specifies the following properties
  #   label: A name for the condition. This is the id/key for the
  #          condition. If this is changed, all previous classifications
  #          will be reset. Likewise, if you change the dimensions but keep
  #          the condition label the same, existing classifications for
```



```
  #          that condition will be retained.
  #   dimensions: A list of dimensions whose intersection forms the IMEI
  #               set result for the condition. Each of these can take
  #               parameters that are particular for the dimension being
  #               used. Additionally, they all accept an 'invert' property,
  #               which basically NOTs the result of the dimension by
  #               taking the all-time observed IMEIs list and subtracting
  #               the set of IMEIs returned by this dimension
  #   grace_period_days: The integer number of days that an IMEI failing
  #     this condition will remain on the notification list before moving
  #     to the black list.
  #   blocking: A boolean stating whether this condition contributes to
  #     list generation or is simply informational. Information conditions
  #     can be used to try out new modules or to tweak parameters.
  #   reason: A string sent to the operators describing why the IMEI is
  #     to be blacklisted.
  #   max_allowed_matching_ratio: The maximum percentage of all-time seen
  #   IMEIs this condition is allowed to match. This is a safety check
  #   implemented to catch a missing GSMA TAC DB, registration list, etc.
  #
  # The following are just sample conditions designed to show the features
  # of DIRBS Core and just an example of simple/compound conditions. They
  # are not supposed to represent suggestions for real business rules.
  # Please consult the release documentation for available dimensions and
  # their parameters.
  - label: simple_dimension
    dimensions:
      - module: gsma_not_found
    grace_period_days: 30
    blocking: true
    reason: Violated simple dimension
    max_allowed_matching_ratio: 0.1
  - label: compound_dimension
    dimensions:
      - module: stolen_list
      - module: duplicate_daily_avg
        parameters:
          threshold: 3.1
          period_days: 30
          min_seen_days: 5
        invert: True
    grace_period_days: 0
    blocking: true
    reason: Violated compound dimension
    max_allowed_matching_ratio: 0.1

# Definition of settings to be used for logging output of DIRBS system.
logging:
```



```
  # Logging level determines the verbosity of logs. This is also set to
  # 'debug' by the -v CLI option
  level: info
  # Format string can be configured here
  format: '%(asctime)s - %(name)s - %(levelname)s - %(message)s'
  # Set this to true if you want to see logging message for StatsD
  show_statsd_messages: False
  # Set this to true if you want to see Werkzeug internal log messages from
  # TAC/IMEI APIs
  show_werkzeug_messages: False
  # Set this to true if you want to see SQL messages from DIRBS (most are
  # debug level)
  show_sql_messages: False
  # If log_directory is set to a value that is not "null", DIRBS will log
  # to a file as well as to the console. The log files will all be
  # generated in the directory specified by this setting. This directory
  # needs to exist and be writable
  log_directory: /var/log/dirbs
  # Uncomment and set this value if you want to prefix all log files
  # created on this host with a prefix to distinguish them from other host
  file_prefix: null
  # Set the number of bytes before a logfile is rotated. If this or
  # file_rotation_backup_count is zero, rotation is disabled
  file_rotation_max_bytes: 100000000
  # Sets the numbwe old logs to keep
  file_rotation_backup_count: 100

# Definition of settings to be used for forwarding application-defined
# metrics to a StatsD server for aggregation
statsd:
  # The hostname for the StatsD server. Overridden by environment
  # variable DIRBS_STATSD_HOST if set.
  #
  # Uncomment this and set to a real StatsD hostname to enable collection
  # of metrics
  # hostname = statsd.local
  #
  # The UDP port that the StatsD server is listening on for metrics.
  # Overridden by environment variable DIRBS_STATSD_PORT
  port: 8125
  # The prefix to be used for for all metrics collected from this instance.
  # This is useful if you have multiple hosts or environments sending data
  # to the same StatsD server and you with to differentiate them,
  # Overridden by the environment variable DIRBS_ENV if set.
  #
  # Uncomment this and set to a prefix string to enable prefixing of StatsD
  # metrics
  # prefix =
```



```
# Definition of settings to be used during data cataloging process.
catalog:
  # The prospector harvests all files in the path adding them to the data
  # catalog.
  # Each prospector specifies the following properties:
  #   file_type: Type of files contained within the specified paths.
  #              It should match the keyword specified during dirbs-import
  #              (eg. operator, gsma_tac etc.)
  #   paths: Directories and/or files to be harvested. Sub-directories
  #          within the listed path are not traversed automatically; they
  #          should be listed separately if files within them need to be
  #          cataloged. Multiple paths can be defined for each file type
  #          and the path used should be absolute and globally unique.
  #   schema_filename: Schema file to be used for data pre-validation (if
  #                    enabled).
  # Multiple prospectors can be defined for the same file_type if files
  # exist across multiple schema versions.
  prospectors:
    - file_type: operator
      paths:
        - /path/to/operator_data/directory
      schema_filename: OperatorImportSchema_v2.csvs
    - file_type: operator
      paths:
        - /path/to/operator_data/directory/operator_data_file
      schema_filename: OperatorImportSchema.csvs
    - file_type: gsma_tac
      paths:
        - /path/to/gsma_tac/directory
      schema_filename: GSMASchema.csvs

  # Set this to true if pre-validation should be performed on the data
  # files.
  # Note: Enabling this can slow down the process if there are a lot of
  # uncataloged files.
  perform_prevalidation: False

  # Definitions of operational modes used by DIRBS core to support dynamic
  # whitelist mode
  operational:
  # Boolean variable used to activate whitelist mode, default False
  # By setting it true will result in activation of extra migrations and
  # functions related to DIRBS Core Whitelisting mode
    activate_whitelist: False
  # The boolean variable to toggle the settings weather to share the
  # whitelist with the operators or not.
    restrict_whitelist: True
```



```
# Definitions of settings to be used for broker integration in the DIRBS
# System. Currently broker is only supported for whitelist mode.
broker:
# Definitions of settings to be used for Apache KAFKA integration and
# used as message broker. It will be used for sending and receiving
# messages and data between EIR(s) and the DIRBS System.
 kafka:
  # The host name for the KAFKA server, overridden by environment
  # variable DIRBS_KAFKA_HOST if set.
      hostname: kafka
  # The available port for the KAFKA server, overridden by environment
  # variable DIRBS_KAFKA_PORT
      port: 9092
  # The topic DIRBS will use to receive notifications and requests from
  # operators, overridden by environment variable DIRBS_KAFKA_TOPIC if
  # set.
      topic: test
# Definitions of EIR(s) with their respective operators and topic names
# to subscribe to
operators:
  - id: operator1
    name: Operator 1
    topic: operator1_topic
  - id: operator2
    name: Operator 2
    topic: operator2_topic
```


B Sample Conditions: YML

Table B7-7 YML sample configuration

Condition module name	Sample configuration	Default config
duplicate_daily_avg	- label: duplicate_daily_avg dimensions: - module: duplicate_daily_avg parameters: threshold: 2.0 period_days: 30 min_seen_days: 2 grace_period_days: 0 blocking: true sticky: false reason: Duplicate daily avg detected	N/A
duplicate_threshold	- label: duplicate_mk1 dimensions: - module: duplicate_threshold parameters: threshold: 5 period_days: 120 grace_period_days: 90 blocking: false reason: Duplicate threshold exceeeded max_allowed_matching_ratio: 0.1	N/A
gsma_not_found	- label: gsma_not_found dimensions:	'RBI':'Delay(days)' '00': 32,

Note: Do not use this condition if there is a live DRS enforcing GSAM not found.	- module: gsma_not_found parameters: ***ignore_rbi_delays: False*** ***per_rbi_delays:*** ***"00": 0*** ***"01": 0*** grace_period_days: 0 blocking: true reason: TAC not found in GSMA TAC database max_allowed_matching_ratio: 0.1 ***Note: "bold italics" indicates optional parameters***	'01': 40, '35': 20, '86': 19, '91': 20, '99': 69 See GSM Association Non Confidential Official Document TS.06 (DG06) IMEI Allocation and Approval Guidelines for additional information on RBIs
malformed_imei	- label: malformed_imei dimensions: - module: malformed_imei grace_period_days: 0 blocking: true reason: Invalid characters detected in IMEI max_allowed_matching_ratio: 0.1	N/A
not_on_registration_list	- label: not_on_registration_list dimensions: - module: not_on_registration_list grace_period_days: 0 blocking: true reason: IMEI not found on local registration list max_allowed_matching_ratio: 1.0	N/A
stolen_list	- label: local_stolen dimensions: - module: stolen_list grace_period_days: 0 blocking: true max_allowed_matching_ratio: 0.1 reason: IMEI found on local stolen list	N/A
used_by_dirbs_subscriber	- label: used_by_local_non_dirbs_roamer	N/A

	dimensions: - module: used_by_dirbs_subscriber parameters: lookback_days: 2 grace_period_days: 0 reason: IMEI found for local non DIRBS roamer	
used_by_international_roamer	- label: used_by_local_non_dirbs_roamer dimensions: - module: used_by_international_roamer parameters: lookback_days: 2 grace_period_days: 0 reason: IMEI found for local non DIRBS roamer	N/A
used_by_local_non_dirbs_roamer	- label: used_by_local_non_dirbs_roamer dimensions: - module: used_by_local_non_dirbs_roamer parameters: lookback_days: 2 grace_period_days: 0 reason: IMEI found for local non DIRBS roamer	N/A
exists_in_barred_list	- label: barred_list dimensions: - module: exists_in_barred_list grace_period_days: 0 blocking: true max_allowed_matching_ratio: 0.1 reason: IMEI found on barred list	N/A
is_barred_tac	- label: barred_tac_imeis dimensions: - module: is_barred_tac grace_period_days: 0 blocking: true max_allowed_matching_ratio: 0.1	N/A

	reason: IMEI belongs to barred tac	
daily_avg_uid	- label: duplicate_daily_avg_uid dimensions: - module: daily_avg_uid parameters: threshold: 2.0 period_days: 30 min_seen_days: 2 grace_period_days: 0 blocking: true sticky: false reason: Duplicate daily avg UIDs detected	N/A
exists_in_monitoring_list	- label: monitored_imeis dimensions: - module: exists_in_monitoring_list grace_period_days: 16 blocking: false max_allowed_matching_ratio: 0.1 reason: IMEI is being monitored for behavior	N/A
not_on_association_list	- label: non_associated_imeis dimensions: - module: not_on_association_list grace_period_days: 15 blocking: true max_allowed_matching_ratio: 0.1 reason: IMEI not registered to UID	N/A

