NetGuardians

RiskTech 100 2018

Gartner 2015 Cool Vendor

Reference Data Administration

September 2019

NG|Academy

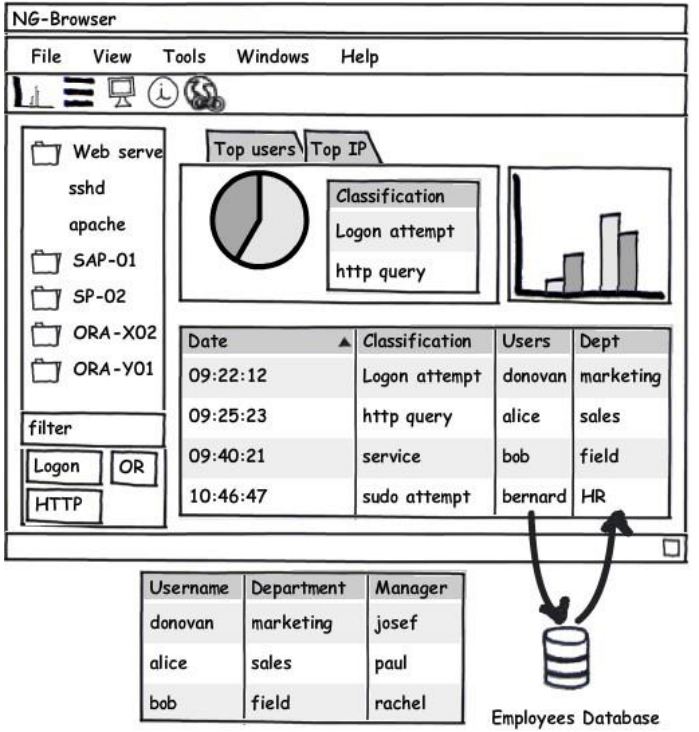
swiss made software

Summary

- Overview
- Process
- Configuration
  - Cache definition
  - Translator definition
- Reload Cache
- Troubleshooting

Overview

- Goal
  - Enrich forensics and controls with additional data
- Method
  - Done at normalization
    - Inserted in NG|Storage
  - Creates Caches (Key → Values)
  - Replace or complete fields



Overview

- External data sources that could be used to populate cache
  - SQL Databases
  - LDAP Directories
  - CSV Files
- Cache with key/values pairs
  - Key → existing field in collected events
- Poll external data sources or read local CSV file
  - Create local cache → Not querying external sources every time

Process

1. Querying external source on reload cache
   - Building local cache

1. On normalization event, reference data will be queried

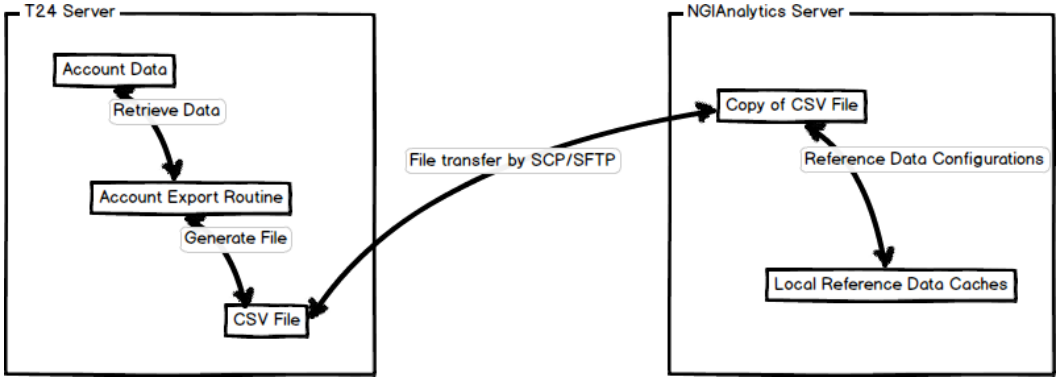
2. If translator for event exists, will look in reference data cache

3. If entry in cache is found, add values to normalized events

Example: T24 Accounts

- Retrieval of data
  - **When T24 on external DB:** Direct poll on Account table
  - T24 Routine extracting data to CSV file. CSV will then be sent to NG|Screener server (cf. automation training)

- Examples
  - Dormant accounts
  - Posting restrictions
  - Account category



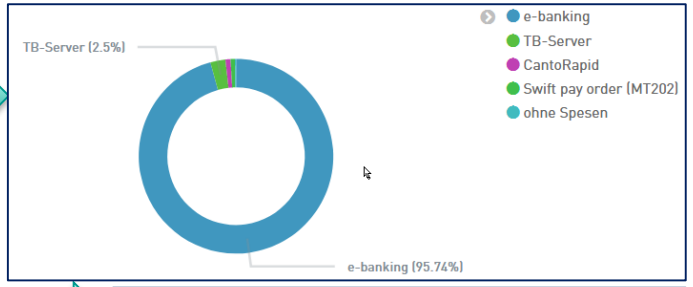
Examples of Reference Data

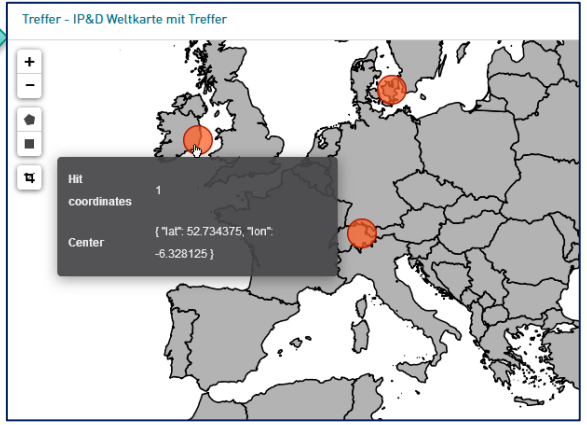
Reference data → Use case - **control** → Use case - **visualization**

channel	110
channel_desc	e-banking

session_source_ip	city_of_connection	coordinates_geo	country
213.55.211.88	Renens	46.5321,6.58832	CH
80.219.26.35	Bern	46.948,7.44745	CH
79.233.191.37	Frankfurt am Main	50.1109,8.68213	DE
178.194.155.189	Zurich (Wiedikon)	47.3619,8.50966	CH
89.217.126.245	Zurich (Zürich (Kreis 11) / Seebach)	47.4137,8.5526	CH
82.136.100.90	Zug	47.1856,8.49067	CH

e.g. Profile e-banking over Bern city





- Reference data:
  - Enrichment – Add a lookup value based on key. e.g. Add city of connection, country of connection and geographical coordinates based on IP
  - Translation – Data capture of the code → translation → have the name associated with the code; **allows java script data manipulation**

Configuration

- Two different configurations
  - Cache definition
    - /etc/ng-screener/common/referenceData/
  - Mapping to Business data model
    - /etc/ng-screener/daemon/modules/feeding/translators/
- XML file format
  - In each case, sample available
    - /etc/ng-screener/common/referenceData/samples/
    - /etc/ng-screener/daemon/modules/feeding/translators/samples/
- Tenant information should be provided (example in the following slides)

Cache definition

- Configuration file composed of three parts
  - Tenant information
  - Data source part
    - Where to search for data
    - CSV, LDAP, Database configuration
  - Cachegroup part
    - Cache definition
    - One cache group per CSV file (or query on database)
    - Query: Select statement to get data for cache creation
    - Cache: one cache definition per key

Translator definition

- Composed of two parts
  - Sources part
    - To which service/host combination to apply caches
    - service@host**@tenant**
    - * wildcard available
  - Translator parts
    - Map key and value from cache to Business data model fields
    - BDM_field_name=Cache_field_name
    - One key tag (possible composite keys)
    - Multiple values
      - Option for values: replace or append
      - Default action: replace

Example: CSV file

- Dormant account example

- `Account.csv` file with
  - `ACCOUNT_NUM` as key
  - `ACCOUNT_NAME` as first value
  - `INACTIVITY_MARKER` as second value

- CSV must contain header used as column name (see figure)

ACCOUNT_NUM	ACCOUNT_NAME	INACTIVITY_MARKER
123456	Acc1	N
234567	Acc2	Y
345678	Acc3	N

Example: CSV source

- Cache configuration example
  - Separator and path defined for CSV file
- In cachegroup
  - Query with SELECT csv_column_name1, csv_column_name2, … FROM csv_filename_without_ext
  - Cache definition as KEY -> VALUE_1, VALUE_2, …, VALUE_N

```
<cacheconfig>
    <tenant>DEFAULT</tenant>
    <datasource type="csv">
        <!--<connectionRetryDelay>0</connectionRetryDelay> -->
        <!--<maxConnections>0</maxConnections> -->
        <!--<fetchSize>50000</fetchSize> -->
        <separator>;</separator>
        <!--<fileExtension>.csv</fileExtension> -->
        <!--<suppressHeaders>false</suppressHeaders> -->
        <!--<headerLine></headerLine> -->
        <!--<skipLeadingDataLines>0</skipLeadingDataLines> -->
        <path>/home/ng-screener/</path>
    </datasource>

    <cachegroup name="ACCOUNT" cacheInMemorySize="50000" cacheRefresh="86400">
        <query>
            select ACCOUNT_NUM, ACCOUNT_NAME, INACTIVITY_MARKER from Account
        </query>
        <cache name="AccountNum_AccountNameInactivityMarker" inMemorySize="30000">
            ACCOUNT_NUM -> ACCOUNT_NAME, INACTIVITY_MARKER
        </cache>
    </cachegroup>
</cacheconfig>
```

Example: CSV source

- Translator configuration example
  - Source: Apply to all host with temenosT24Transaction service
  - Translator
    - Key: map ACCOUNT_NUM in CSV to transaction_sender_account_id in BDM
    - Values: BDM_Field=CSV_Field (BDM could be extended)

```
<translatorconfig>
    <sources>
        <source>temenosT24Protocol@*@DEFAULT</source>
    </sources>

    <translator>
        <key>transaction_sender_account_id=ACCOUNT_NUM</key>
        <value>transaction_sender_account_name=ACCOUNT_NAME</value>
        <value action="replace">transaction_sender_account_dormant_flag=INACTIVITY_MARKER</value>
    </translator>

</translatorconfig>
```

Example: Historical values

- If controls are executed in the past, reference data could not be relevant anymore
  - Reference data relevant for current day NOT necessarily for past days

- Timestamp of reference data should be added
  - Reference data could be defined on composite key
    - ID + Timestamp
    - Timestamp should not have to match → Upper or lower bounds could be specified

Example: Historical values

- Cache definition part

```
<cachegroup name="FXRate_hist" cacheSize="50000" cacheRefresh="86400">
    <query>
        select BASE_CCY as FX_BASE_CCY, DATE as FX_DATE, RATE as FX_RATE from fxrates
    </query>
    <cache name="FXRate" keyMatch="exact, lower" keyClass="string, date" keyFormat=", 'dd.MM.yyyy'" size="30000">
        FX_BASE_CCY, FX_DATE -> FX_RATE
    </cache>
</cachegroup>
```

Matching on the keys. Provided in the format “first key, second key, …”
In this example, we have an exact match on FX_BASE_CCY. “lower” means we will find the closest date up to the date in the data.

Format of the different keys using the same format as for keyMatch. In this example FX_BASE_CCY is a string and FX_DATE is a date.

For specific types (keyClass parameter) a format has to be defined. It is the case for date format. In this example, the date that we have in FX_DATE as the format ‘dd.MM.yyyy’.

Example: Historical values

- Translator format

```
<translator>
    <key>transaction_sender_currency=FX_BASE_CCY</key>
    <key class="date" format="dd.MM.yyyy">transaction_date=FX_DATE</key>
    <value>transaction_exchange_rate=FX_RATE</value>
</translator>
```

Both key are defined. In the example the FX_BASE_CCY key will be found in `transaction_sender_currency` in BDM. FX_DATE will be found in `transaction_date`

As for the cache definition, the type (class) and format for the date should be defined.
Note: In the cache we define the format of the date as it is in the cache. Here we define it as it is in the date in BDM.

Scripted Fields

Description

- Make transformations on/using existing fields
- JavaScript language
- Evaluated last
  - Can use fields that are coming from Reference data itself

Configuration

- In translator configuration
- Add `<scriptedField>` tag
- Composed of
  - `<field>:` field to be added or overwritten
  - `<script>:` script to be executed to apply the transformation

Examples of scripted fields usage

From timestamp obtain:

- **Part of day**
- **Day of week**

Translate unexpected input values:

- **Obtain country name from city names: e.g. MIES-TANNAY → CH; Mauren FL → LI**

Replace comma with point inside amounts retrieved as text

Extract ISO-2 letter country code from:

- **SWIFT/BIC: characters 5&6: e.g. UBSWCHZH80A → CH**
- **IBAN: characters 1&2: e.g. RO54BRDE360... → RO**

Example: Scripted fields (1/2)

```
<!-- Sample to overwrite an existing field -->
<scriptedField overwrite="true">
    <field>existing_field</field>
    <script>event['business_reference'] + '_HardCodedString'</script>
</scriptedField>

<!-- Sample to add a new field. Nothing done if the field already exists -->
<scriptedField>
    <field>day_of_week</field>
    <script>["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"][new Date(event['@timestamp']).getDay()]</script>
</scriptedField>

<!-- Samples of string processing -->
    <scriptedField>
        <field>transaction_receiver_bank</field>
        <script>event['transaction_receiver_branch_id'] ? (event['transaction_receiver_branch_id'] || '').trim().split(" ").slice(-1)[0] : 'Unknown'
        </script>
    </scriptedField>
    <scriptedField>
      <field>transaction_receiver_bank_reference</field>
      <script>(event['transaction_receiver_bank_id'] || '').trim().split(" ").slice(0, -1).join(" ").trim()
      </script>
    </scriptedField>
    <scriptedField overwrite="true">
        <field>transaction_receiver_bank_country</field>
        <script>event['transaction_receiver_branch_id'] ? (event['transaction_receiver_branch_id'].trim().split(" ").slice(-1)[0].substring(4,6)) :
        (event['transaction_receiver_account_id'] ? ((match = (event['transaction_receiver_account_id'] ||
        '').trim().split(/\s/).slice(-1)[0].match(/([A-Z]{2,3})[0-9]+/)) ? match[1] : (event['transaction_receiver_account_id'] ||
        '').trim().split(/\s/).slice(-1)[0].substring(4,6)) : 'Unknown')
        </script>
    </scriptedField>
```

Example: Scripted fields (2/2)

```
<!-- Sample of replacing , with . in amount -->
    <scriptedField overwrite="true">
      <field>transaction_sender_charges_amount</field>
      <script>(event['transaction_sender_charges_amount'] || '').trim().replace(/,$/,'').replace(',', '.')
      </script>
    </scriptedField>

<!-- Sample for adding normalization time and date -->
   <scriptedField>
       <field>normalization_date_trx_pillars</field>
       <script>new Date().toLocaleString()
       </script>
    </scriptedField>
<!--Sample for adding a field with a fixed value - useful to identify new customers -->
    <scriptedField>
       <field>new_object_identifier</field>
       <script>"NC"
       </script>
    </scriptedField>
<!--Sample of function -->
    <scriptedField overwrite="true">
      <field>receiver_bank_id_desc</field>
      <script>(function(){
          var bank_type = event['receiver_bank_type'],
          receiver_bank_id_desc;

          if (bank_type.indexOf('BP') !== -1 ) {
            receiver_bank_id_desc=event['receiver_bank_id_desc'];}
          else {
              receiver_bank_id_desc=event['receiver_bank_id'];
          }
          return (receiver_bank_id_desc || '');
        })()
      </script>
    </scriptedField>
```

Reload Cache

- To reflect changes happening on external data sources to local cache, reload of them is needed

- Two ways to reload
  - Using `ngadmin` command line tool
    - `ngadmin –tenant=TENANT_NAME referencedata_reloadCaches [-g | --group] [--clear] [-h | --help]`
  - Specify `cacheRefresh` property on `cachegroup` tag

- Preferred method: `ngadmin` command with an entry in crontab

- When configuration has changed (either cache or translator*), ng-screener has to be restarted
  - `sudo systemctl restart ng-screener`

- *For only translator modifications, ng-screener restart is not mandatory, it can be replaced with a faster command:
  - `ngadmin reload`

Useful commands for verification

- `ngadmin referencedata_listCaches –a`
  - List caches state and size (-a)

```
                          ]# ngadmin referencedata_listCaches -a

    NAME                                TYPE   STATE    SIZE      CACHE
------------------------------------  -----  -------  --------  --------------------
channel_desc                          EXACT  LOADED   46        channel -> channel_desc
collective_order_status_desc          EXACT  LOADED   35        collective_order_status -> collective_order_status_desc
collective_order_type_desc            EXACT  LOADED   23        collective_order_type -> collective_order_type_desc
```

- `ngadmin referencedata_listCacheEntries -k channel -l 50`
  - Using key (-k) channel, display the first 50 lines of this cache (-l)

```
                          # ngadmin referencedata_listCacheEntries -k channel -l 50

 channel |       channel_desc
---------+--------------------------
 12      | EBS
 13      | Reuters System
 14      | Reuters2000
 15      | DTA
 2002    | !EBES EGA-B Giro
 2001    | !EBES EGA-B Cash
 2023    | SEPA Direct Debit Repair
 2000    | CantoRapid
 2022    | SEPA Direct Debit
```

- `ngadmin referencedata_listCacheEntries -k channel -s 13`
  - Displays the value from *channel* for the key 13

```
                          # ngadmin referencedata_listCacheEntries -k channel -s 13

 channel |  channel_desc
---------+----------------
 13      | Reuters System
(1 entry)
```

Troubleshooting

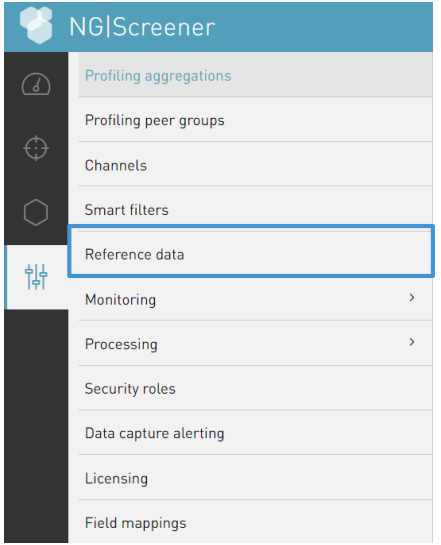
- If problem when reloading cache, use verbose mode
  - `sudo ngadmin -v referencedata_reloadCaches`
- Force reloading of caches
  - `sudo ngadmin referencedata_reloadCaches --clear`
- Check application logs
  - `tail -f /var/log/ng-screener/daemon/daemon-referencedata.log`
  - `tail -f /var/log/ng-screener/daemon/daemon-all.log`
- Reference data caches are stored in NG|Storage

Troubleshooting

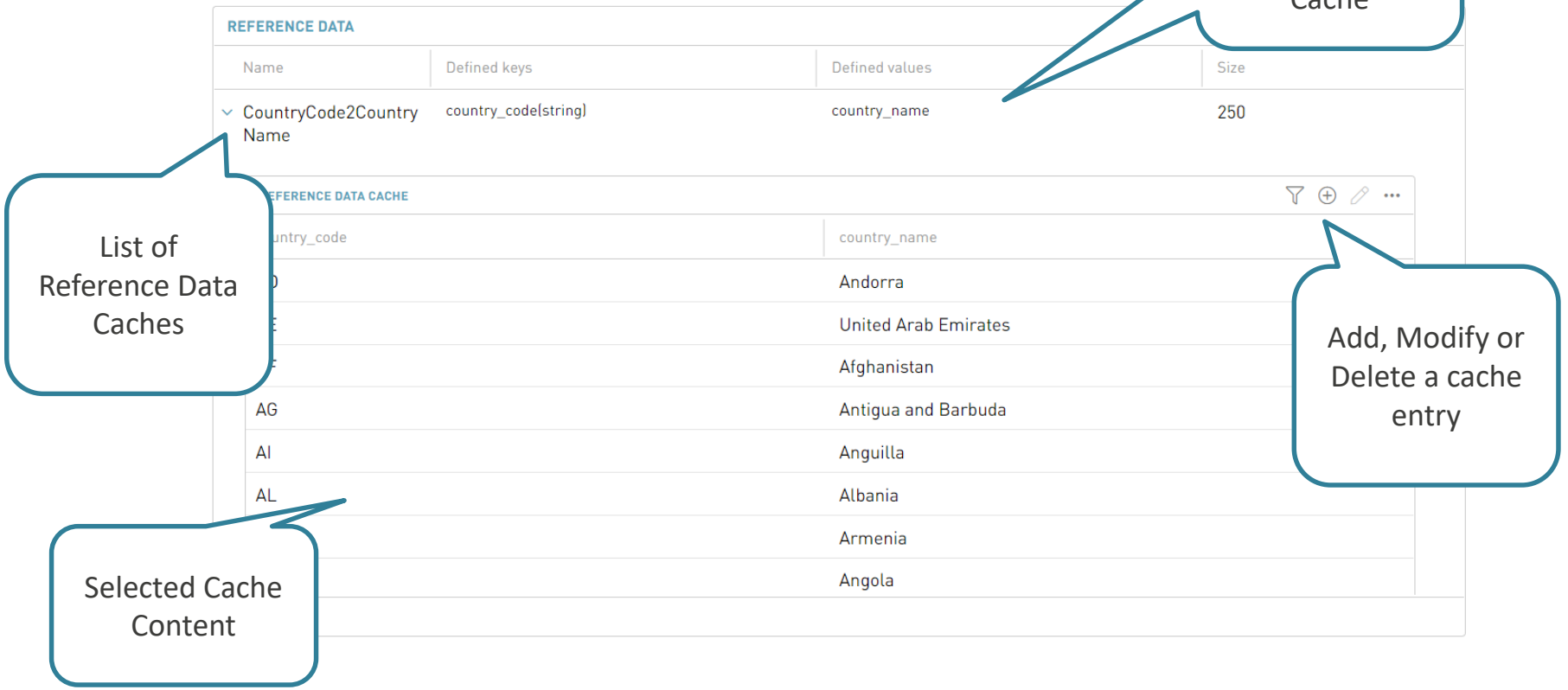
- If reference data is not present in NG|Storage, check the following
  - `ng-screener` service has been restarted
  - Caches have been successfully reloaded (no errors in log files)
  - Data in NG|Storage has been reloaded
    - `data_removeEntries`
    - `data_launchInitialProcessing`

Reference Data in NG|Screener UI

- In admin part of NG|Screener UI
- Can visualize cache defined
- New entries can be added from there
  - Whitelist or blacklist usage for example



Reference Data in NG|Screener UI



THANK YOU!

Contact us

+41 24 425 97 60

info@netguardians.ch

www.netguardians.ch

Linkedin.com/company/netguardians

Facebook.com/NetGuardians

@netguardians

https://www.youtube.com/netguardians

NetGuardians Headquarters

Y-Parc, Av. des Sciences 13
1400 Yverdon-les-Bains
Switzerland

T +41 24 425 97 60
F +41 24 425 97 65

NetGuardians Africa

KMA Centre , 7th floor,
Mara Road Upper Hill,
Nairobi, Kenya

T +254 204 93 11 96

NetGuardians Asia

143 Cecil Street
#09-01 GB Building
069542 Singapore

T +65 6224 0987

NetGuardians
Eastern Europe

Koszykowa 61, 00-667
Warsaw, Poland

NetGuardians Germany

Rhein-Main Gebiet
Germany

T +49 172 3799003