# Specification for daily harvesting of data from the BDR

Compared to the pre-2014 architecture, there is more data exchange between the BDR and the database. New types of data that are to be pulled from BDR are:

* company data, handed over to BDR from the F-gas portal
* metadata for BDR envelopes that have been rejected
* metadata for draft BDR envelopes that have not yet been submitted

Additionally, the reporting framework has changed considerably such that the retrieval of BDR accepted envelopes is more complex than before.

# Harvesting of company data from the portal/BDR admin

## Framework

Company Data from the portal are regularly (every 30 minutes) copied from the F-gas portal into the BDR admin database.

This data set includes **per company**:

1. **Company data (once):**

|  |  |  |
| --- | --- | --- |
| All Companies | EU Companies | Non-EU Companies |
| Name | VAT No. | OR name |
| Address (four fields) |  | OR address (four fields) |
| Telephone |  | OR telephone |
| Website |  | OR website |
| Fgas12345 number\* |  | OR VAT No. |
| Business profile (fifteen boolean fields)\*\* |  | OR first name of person |
|  |  | OR last name of person |
|  |  | OR e-mail |

\* Only for companies registered in the BDR before mid-2014.

\*\* a series of seven generic yes/no statements on certain activities; two specific subfields for equipment importers; six specific subfields for recipients of exempted HFCs.

1. **Multiple user records in no particular order:** first name; last name; e-mail (Note that these data are independent from and in addition to the one person who is stored along with OR company data in the company record itself.)
2. **Note on business profile:** Because of the different structures in circulation, we accept some redundancy and store both parent fields (importer as well as specific import types) and child fields.

Furthermore the BDR admin will continue to hold old company & contact person data related to companies which did not (yet) register in the Portal.

## Required data for harvesting

* Copy & store in ‘Fgases aggregation.database’ **all** data sets related to companies holding a Portal ID code
* Copy & store in aggregation database all user accounts related to those companies
* Do **not** copy data related to companies which have only a BDR\_username-ID and no Portal code.

## Importation procedure into FDB

Importation must make sure that changed data are reflected, historic data is retained, and no account is doubled. **Note that due to the multitude of identification schemes that keep changing, the FDB uses an internal ID for all referencing of companies, which is saved in the ID\_internal field of tbl\_companies\_map and mapped there to all other types of code that are or have been in circulation.**

There are three related tables in FDB:

* tbl\_companies\_map maps internally used IDs to portal codes and other identification codes
* tbl\_companies\_data holds current and historic company data. The current record is referenced by valid\_dataset in tbl\_companies\_map.
* tbl\_users holds current and historic user accounts for companies. Current records have field valid=true.

### Verify existing company data

* If portal code is 10086 (dummy testing company), proceed with next, do not import.
* For each company, check whether portal code is present in tbl\_companies\_map, and note ID\_internal field if record is found.
* **If portal code does not exist** in FDB: See if harvested data contains a BDR username (“fgas12345”).
  + If data includes BDR username:
    - Find that BDR username in tbl\_companies\_map. All fgas12345-style usernames should be present there, and the portal code of that record should be empty, if not, something went wrong: please notify FGas team.
    - set portal\_code to of that record to the harvested portal code.
    - Proceed with comparison of company data as described below for existing companies.
  + If data does not include BDR username:
    - Create new record in tbl\_companies\_map, write portal code and Country\_ID, leave empty other fields, and note resulting autonumber ID\_internal.
    - Change date\_fdb\_registration only if needed, it will default to current date and time.
    - Create new record in tbl\_companies\_data, set Company\_ID to the new internal ID, write data according to DB documentation.
* **If portal code exists or was mapped to an existing company in the previous step**: Query tbl\_companies\_map for the valid\_dataset field of the Company\_ID. Look for that dataset\_ID in tbl\_companies\_data. Compare all fields of resulting record with the newly harvested company data. In case of any change, a new record will be created and the old data kept for reference.
  + If a difference is found:
    - create new record with harvested company data including internal Company\_ID which is used to identify historical records. It is timestamped automatically by FDB.
    - Set valid\_dataset in tbl\_companies\_map to Dataset\_ID of newly created dataset.
  + If no difference is found: No action necessary, data is current.

### Verify list of users

* For existing companies, query tbl\_users for Company\_ID = ID\_internal (may yield multiple records)
* Compare resulting records to newly harvested user list. Users shall be treated as existing if all fields (first name, last name, e-mail) match. There is no adjustment of old records, only creation of new ones, even for changes.
* If a user exists in FDB, but not in harvested recordset, they were probably removed or changed. Set that record’s valid flag to FALSE in FDB, proceed with next.
* If a user exists in harvested recordset, but not in FDB, they are probably new or have been changed. Create new record in tbl\_users, write data and set correct Company\_ID (ID\_internal – not portal code).

# Harvesting of BDR-accepted submissions from the BDR

## Required data

The harvesting procedure shall retrieve full data for all BDR-accepted reports submitted during the present calendar year, that is, all BDR-accepted reports submitted on or after 1 January of the present calendar year. The data can be found in the XML files attached to the BDR envelopes.

BDR accepted envelopes contain four types of data which shall be saved in FDB:

* metadata about the report (required: submission date, BDR delivery URL)
* data about the reporting company along with their portal code, identical to their current portal record on the time of submission, users, and reported affiliations
* data about F-gas mixtures defined by the company
* transaction year and values reported by company

## Data retrieval from BDR

1. Retrieve from BDR the XML files from all F-gas related BDR-accepted envelopes that have been submitted after the current submission date threshold (usually 1 January of the present calendar year).
2. Store full contents in the F-gas aggregation DB.

## Importation procedure into FDB

Each harvested report shall be treated in turn as follows. The order is important: start with the oldest report according to BDR submission date.

**Note that before import of reported data the updated company data should have been imported (section )!**

### Check for existing copy of harvested report

* Determine ID\_internal of the company by querying tbl\_companies\_map for the harvested portal\_code.
* Query tbl\_report for existing reports with company\_ID in question and BDR\_submission\_date equal to the exact submission date and time of the harvested report. If a match is found, abort importation and continue with next harvested report.

### Adjustment of existing reports according to transaction year and submission date of harvested report

This step is **crucial for the correct processing of submitted reports**. Depending on the date of submission and the transaction year provided in the report, there a different requirements for setting the Most\_recent\_report and Status\_EC\_approval\_pending flags of the new report record.

1. Transaction year is *equal to or later than* the year before the current calendar year, and submission date is *no later than* 30 Jun of the current calendar year. Example: Submission of a report for the 2014 transaction year on 22 April 2015.
   * Check for existing reporty by querying tbl\_report for the matching Company\_ID and transaction\_year fields where Most\_recent\_report = TRUE
   * If found: Set the existing record’s Most\_recent\_report flag to FALSE
   * In the record for the new report, set Most\_recent\_report to TRUE and let Status\_EC\_approval\_pending default to FALSE.
2. Transaction year is *equal to or later than* the year before the current calendar year, and submission date is *on or after* 1 Jul of the current calendar year. Example: Submission of a report for the 2014 transaction year on 3 Jul 2015.
   * If the new report is a data report:
     + No querying for existing reports is necessary as they will be matched manually.
     + In the record for the new report, set Most\_recent\_report to FALSE and Status\_EC\_approval\_pending to TRUE.
   * If the new report is a NIL report:
     + Query database for existing reports for this company and the Transaction\_year in question where Report\_Status\_BDR=accepted.
     + If matching reports are found, set Most\_recent\_report to FALSE and Status\_EC\_approval\_pending to TRUE.
     + If no matching reports are found (i. e. this NIL report is the first entry for the transaction year), set Most\_recent\_report to TRUE and Status\_EC\_approval\_pending to FALSE.
3. Transaction year is *more than one year before* the current calendar year. Example: Submission of a report for the 2012 transaction year at any date in 2015.
   * If the new report is a data report:
     + No querying for existing reports is necessary as they will be matched manually.
     + In the record for the new report, set Most\_recent\_report to FALSE and Status\_EC\_approval\_pending to TRUE.
   * If the new report is a NIL report:
     + Query database for existing reports for this company and the Transaction\_year in question where Report\_Status\_BDR=accepted.
     + If matching reports are found, set Most\_recent\_report to FALSE and Status\_EC\_approval\_pending to TRUE.
     + If no matching reports are found (i. e. this NIL report is the first entry for the transaction year), set Most\_recent\_report to TRUE and Status\_EC\_approval\_pending to FALSE.

### Storage of newly harvested report

Create new record in tbl\_report. Most fields default to correct values. Set the following:

* Report\_submission\_no to new unique ID according to scheme to be discussed
* Report\_submission\_date to date and time of submission in BDR
* Transaction\_year to transaction year as provided by report (that is, not the transaction year from the BDR envelope!)
* Most\_recent\_report and Status\_EC\_approval\_pending according to the result of step
* bdr\_delivery\_url to full URL of envelope in BDR
* report\_import\_date\_to\_FDB defaults to current date and time, change only if necessary.
* Report\_status\_BDR to ‘accepted’

### Snapshot of company data in tbl\_report

Company data is stored in tbl\_report along with the report instead of referencing it in order to create a snapshot of valid company data at the time of report creation. The following fields must be filled in the new report record:

1. **For EU companies:**
   * Company\_name, Company\_street, Company\_House\_no, Company\_postcode, Company\_city, Company\_VAT\_No: All of them are strings, set to harvested values
2. **For non-EU companies:**
   * Company\_name, Company\_street, Company\_house\_no, Company\_postcode, Company\_city as above
   * Data for EU representative company goes to OR\_Name, OR\_Street, OR\_House\_no, OR\_postcode, OR\_city, OR\_website, OR\_VAT\_No; data for the person stored with it in the company data to OR\_firstname, OR\_lastname, OR\_email, OR\_telephone
   * Two-letter country code for EU representative goes to OR\_Country\_ID

### Snapshot of company user data in tbl\_report

For each user (company contact) in the reporting XML, add one entry in tbl\_users regardless whether that user already exists or not. However, leave the company\_ID field **empty** and insert the correct reference in **Report\_submission\_no**.

### Snapshot of company business profile in tbl\_actor\_2015

For compatibility reasons, the fields reflecting the company’s business profile as reported are stored in a distinct table, tbl\_actor\_2015, with a 1:1 relationship to tbl\_report. In that table, insert one new record for each harvested report, setting the correct Report\_submission\_no and the flags corresponding to the business profile included in the report XML.

### Storage of company affiliations in tbl\_company\_affiliations

Company affiliations are reported and stored on a report-by-report basis. Therefore these records refer to reports and not to companies.

Importation is straightforward. For every reported affiliation

* look up the internal ID of the reported affiliate. This can be done via a query on VAT\_no in tbl\_companies\_data (where there exists a record in tbl\_companies\_map whose valid\_dataset references the dataset\_ID of that same data record - should yield only one result).
* Alternatively, if a portal code is provided, the internal ID can be looked up in tbl\_companies\_map.

When the ID is retrieved, create one record in tbl\_affiliations, setting the following fields:

* report\_submission\_no to the report which is being processed (thus also identifying the reporting company)
* affiliate\_company\_ID to the ID\_internal of the affiliate company reported
* additional data as found in report in (technically redundant, but we want to be able to see what was written in a report)
  + CompanyName
  + EUBased
  + EUVAT
  + NonEUCountry
  + NonEUDgClimaRegCode
  + NonEURepName
  + NonEURepVAT

If the affiliate company cannot be found in company data, something went wrong as that report should never have passed QC in the first place. Please notify F-gas team.

### Store user-defined blends

All user defined blends provided in this record shall be stored in FDB and checked manually. For each such blend, do the following:

1. Reporting in 2016 and beyond: check for existence of this exact composition in FDB (method to be detailed)
2. **Create new record in tbl\_gases**, setting the following fields:
   * **sortkey** to 400
   * **custom** and **blend** to TRUE
   * **reported\_trade\_name** to “New mixture: “ & trade name of gas as reported (this is redundant in order to very clearly mark user reported mixtures for all users of the database)
   * **reported\_in** to report\_submission\_no of report which this gas was reported for.
3. **Store composition data in tbl\_blendcomposition**
   * Create one record per component, setting the following fields:
     1. Gas\_ID to ID of new mixture
     2. Component\_ID to ID of component
     3. Percentage to that component’s share in the mixture (float 0–1)
     4. **Only for “other” components**: reported\_other\_explanation to description of “other” component as provided by reporter.
4. **Have FDB generate a composition string as a name**

**Depending on what’s easier to program, either**

* + Query “qry\_gas\_compositions” for Gas\_ID (autonumber) of new gas
  + Set gas\_name\_display of new tbl\_gases record to “Mix: “ & qry\_gas\_compositions.Composition

**OR**

* + Run GetCompositionString(int) function of FDB with Gas\_ID (autonumber) as argument, it will return a string
  + Set gas\_name\_display of new tbl\_gases record to “Mix: “ & [result of function]

1. **Create QC tasks related to new custom blends**

These records in tbl\_qc\_tasks are linked to multiple elements to facilitate finding open questions related to them.

* + All custom blends: Create one record in tbl\_qc\_tasks. Set the following fields
    1. gas\_ID to ID of new blend
    2. Report\_submission\_no to the report where it was defined
    3. type\_ID to 200 (refers to checking composition of a new blend)
  + Custom blends with “other” components: Create another record in tbl\_qc\_tasks. Set
    1. gas\_ID to ID of new blend
    2. Report\_submission\_no to the report where it was defined
    3. type\_ID to 81 (checking of “other” components for validity)

### Store and validate reported values

Store each **reported value**, that is, those entered directly by the reporting company. **Calculated values** will not be stored save the specific charges of gases in products in sheet 11 (see step 2.3.10).

For each reported value, create one record in tbl\_value, setting the following fields:

* **Report\_submission\_no** to ID of report.
* **Transaction\_ID** to **numerical code** of transaction in question. This is **not** the 2E, 2A, … code from the form, which is stored in code\_2015 of tbl\_transaction, but the internal ID which is stored in Transaction\_ID of tbl\_transaction! If unsure, query tbl\_transaction for the new “2E” style code (field code\_2015) to retrieve the matching Transaction\_ID.
* **Gas\_ID**: If this value refers to a particular gas, set this to ID of gas, mixture or custom mixture for which transaction is reported
  + **Exception:** reported amounts of product in section 11 do not relate to a particular gas; they receive a transaction ID, but not a Gas ID
* **For transactions with an EU trade partner specified: (list goes here)**
  + trade\_partner\_ID to ID\_internal of partner company (not necessary for 5C\_voluntary)
  + trade\_partner\_name to name as provided
  + trade\_partner\_EU defaults to TRUE
  + trade\_partner\_EU\_VAT\_no to VAT No. as specified
* **For transactions with a non-EU trade partner specified:**
  + trade\_partner\_ID to ID\_internal of partner company (not necessary for 5C\_voluntary)
  + trade\_partner\_name to name as provided
  + trade\_partner\_EU to FALSE
  + trade\_partner\_nonEU\_OR\_name to name of OR as provided
  + trade\_partner\_nonEU\_OR\_VAT\_no to VAT No. of OR as provided
  + trade\_partner\_nonEU\_portal\_code to ECAS portal code of trade partner, if available
  + trade\_partner\_nonEU\_country to two-letter code of trade partner country as provided
* **For transactions with a voluntary or mandatory explanation attached:**
  + Use description\_reporter to store explanations provided by company.
  + Use description\_reporter\_transaction for section 11, where there may be an explanation provided for a whole product category which would thus apply to all values reported for this product category. Store these explanations in description\_reporter\_transaction of all the related values. The transactions where this applies are: 11A3, 11A9, 11A12, 11B3, 11B5, 11B7, 11B9, 11D1-3, 11E4, 11F9, 11H1-4, 11P.
* **num\_value** to numerical value as reported
* **For values in a unit other than metric tonnes**
  + **unit\_name\_short** to the correct unit (reference: listed in tbl\_units)
* **If any QC checks related to this value have resulted in “flag for manual QC”**
  + For each QC routine that has resulted in “flag for manual QC”, create a record in tbl\_qc\_tasks, setting the following fields:
    1. value\_ID to autonumber of the value in question
    2. report\_submission\_no to report in question
    3. type\_ID to code of necessary check, as defined by QC scheme and tbl\_qc\_task\_types
  + Note that several QC tasks may need to be set related to the same value.

### Storage of calculated values where necessary

In addition to reported values, some **complex calculated values** are stored with the report because they may have additional information attached to them.

* specific charges determined in section 11: store all of them, even if no comment attached, with appropriate unit and transaction ID, providing the explanation in description\_reporter.
* For additional clarity set gas ID of those specific charges to generic “F-Gases” ID 142.
* Set the calculated flag of those transactions to TRUE.

### Check consistency of reported stocks

Quality control shall ensure that reported stocks are consistent across transaction years. The importation procedure must therefore flag any gases missing from the current report where there were end-of-year stocks reported for the previous year.

* Query tbl\_report for a report of company in question with Most\_recent\_report = TRUE for the year *preceding* the transaction year of the harvested report. You need to select the Report\_submission\_no and scheme\_version fields.
* **If a matching report exists**, continue by looking for reported 31 Dec stocks in it depending on the existing report’s scheme\_version.
  + For **scheme\_version = 1**: Query tbl\_value for values related to the report found with **transaction\_ID 9** (stocks held 31 Dec).
  + For **scheme\_version = 2:** Query tbl\_value for values related to the report found with **transaction\_ID 227** (4F – total 31 Dec stocks).
* For each record returned, check whether a corresponding value with the same Gas\_ID for 1 Jan stocks (transaction ID 226) has been included in the newly harvested report. If there is none found:
  + create a record in tbl\_value for each such gas, setting the following fields:
    - report\_submission\_no to newly harvested report,
    - Gas\_ID to Gas\_ID where there was no match in the new report
    - transaction\_ID to 226 (1 Jan total stocks),
    - num\_value to 0
  + create a recordin tbl\_qc\_tasks, setting the following fields
    - value\_ID to the autonumber ID of the value which was just created,
    - type\_ID to 42 (missing gas for stock consistency
* **If no matching report exists**, no need for further consistency checks. Continue.

# Harvesting of BDR-rejected submissions from the BDR

## Required data

The harvesting procedure shall retrieve meta and company data for all BDR-rejected reports submitted during the present calendar year, that is, all BDR-rejected reports submitted on or after 1 January of the present calendar year.

BDR rejected envelopes feature two types of data which shall be saved in FDB:

* metadata about the report (required: submission date and time, BDR delivery URL)
* Portal ID of reporting company

## Import procedure

**Note that before import of envelope data the updated company data should have been imported (section )!**

### Check for existing copy of current report

Query tbl\_company\_map for the ID\_internal according to the company’s portal\_code as provided by BDR.

Query tbl\_report for a report with that Company\_ID and BDR\_submission\_date equal to the submission date and time of the harvested report. If there is a report found, abort importation of this report and continue with next report.

### Create record for new rejected report

Because rejected submissions will never receive the “Most\_recent\_report” flag, we do not have to check for existing reports where that flag may be set to true. Therefore, importation is straightforward:

Create new record in tbl\_report. Most fields **default** to correct values. Set the following:

1. Report\_submission\_no to new unique ID according to agreed scheme
2. Report\_submission\_date to date and time of submission in BDR
3. Report\_Status\_BDR to “rejected”
4. Company\_ID to ID\_internal of company
5. bdr\_delivery\_url to full URL of envelope on BDR
6. report\_import\_date\_to\_FDB defaults to current date and time, change only if necessary.

# Harvesting of draft BDR envelopes from the BDR

## Required data

The harvesting procedure shall retrieve meta and company data for all draft BDR-envelopes created or last modified during the present calendar year. Because there is no submission date, the **date and time of last modification** on the BDR are needed.

Draft BDR envelopes feature two types of data which shall be saved in FDB:

* metadata about the report (required: date and time of last modification, URL of envelope in BDR)
* the portal ID of the reporting company

## Importation procedure

**Note that before import of envelope data the updated company data should have be imported (section )!**

### Check for existing copy of current report

Query tbl\_company\_map for the ID\_internal according to the company’s portal\_code as provided by BDR.

Query tbl\_report\_drafts for a report with company\_ID in question and BDR\_draft\_date equal to the date and time of last modification of the harvested draft report.

If there is a report found, no further steps are necessary: continue with next report.

### Create record for new draft report

Create new record in **tbl\_report\_drafts**. Most fields **default** to correct values. Set the following:

* + Report\_draft\_date to date of last modification in BDR
  + Company\_ID to ID\_internal of company
  + bdr\_delivery\_url to full URL of envelope in BDR
  + report\_import\_date\_to\_FDB defaults to current date and time, change only if necessary.