Contactlab DataExchange
Configuration Guide

© Copyright 2026 Contactlab
This document may not be modified, adapted, or reused, in whole or in part, without the prior written permission of the copyright holder.

Summary

1. Introduction ... 3
2. The DataUpdater module ... 4
2.1 Supported file format and datasource structure ... 4
2.2 Location of the datasource ... 6
2.3 Operation modes ... 6
2.4 Additional tools and configuration options ... 7
2.5 Execution and reporting options ... 8
3. The DataImporter module ... 9
3.1 Supported file formats and datasource structure ... 9
3.2 Location of the datasource ... 10
3.3 Operation modes ... 10
3.4 Limitations ... 10
3.5 Additional tools and configuration options ... 10
3.6 Execution and reporting options ... 11
4. The DataCollector module ... 12
4.1 Supported file formats and datasource structure ... 12
4.2 Access and location of the datasource ... 12
4.3 Additional tools and configuration options ... 12
4.4 Execution and reporting options ... 13
5. The DataExporter module ... 14
5.1 Supported file formats ... 14
5.2 Location of the destination ... 14
5.3 Operations mode and available information ... 15
5.3.1 Event info details for email messages ... 17
5.3.2 Event info details for SMS messages ... 18
5.3.3 Device Identification Data for “view” (V) events ... 19
5.3.4 Blacklist identification data in ‘blacklist match’ (K) events ... 19
5.4 XML file format specifications ... 20
5.4.1 Delivery Dataset ... 20
5.4.2 Delivery Info Dataset ... 20
5.4.3 Event Info Dataset ... 20
5.5 Execution and reporting options ... 21
5.6 Data subject to variation over time ... 21
6. The DataDisplacer module ... 22
6.1 Supported file formats and destination structure ... 22
6.2 Location of the destination ... 23
6.3 Additional tools and configuration options ... 23
6.4 Execution and reporting options ... 23
7. Managing DataExchange modules in Contactlab UI ... 24
8. Standard ETL function library ... 26
8.1 Basic::stringDateFormat ... 26
8.2 Basic::getDomainFromEmailAddress ... 26
8.3 Basic::getIntFromBoolean ... 26
8.4 Basic::trimChars ... 26
8.5 Basic::force ... 26
8.6 Basic::integerToBinaryString ... 26
8.7 Basic::semicolonSeparated2bitmask ... 26
8.8 FieldValueMantainer::dontChangeIFFieldValueIS ... 26
8.9 FieldValueMaintainer::genericTranscode ... 26
8.10LookupTable::checkField ... 26
8.11Zone::miscellanea ... 27
9. DataExchange Post-Actions ... 27
9.1 Chained DataExchange Modules ... 27
9.2 Start a Delivery after a DataExchange Operation ... 27
9.3 DB Fitness ... 27
9.4 Duplicate Remover ... 27
10. Diagnostics messages ... 28
11. Frequently Asked Questions & Answers ... 30
12. Contacts ... 31

1. Introduction

The setup of a messaging campaign in Send involves as a prerequisite the setup of a database containing at least all recipients of the campaign. This database has to be setup inside Send.

Send supports a number of alternatives for this task, trying to accommodate the requirements of customers by allowing the configuration of completely automated and safe ways of transferring a large number of recipients and related data as a viable alternative to simple upload operations via the web UI.

Using DataExchange, it is possible to automatically load or update databases in preparation for future campaign deliveries, and also automatically download standard or customized reporting datasets for customers' consumption.

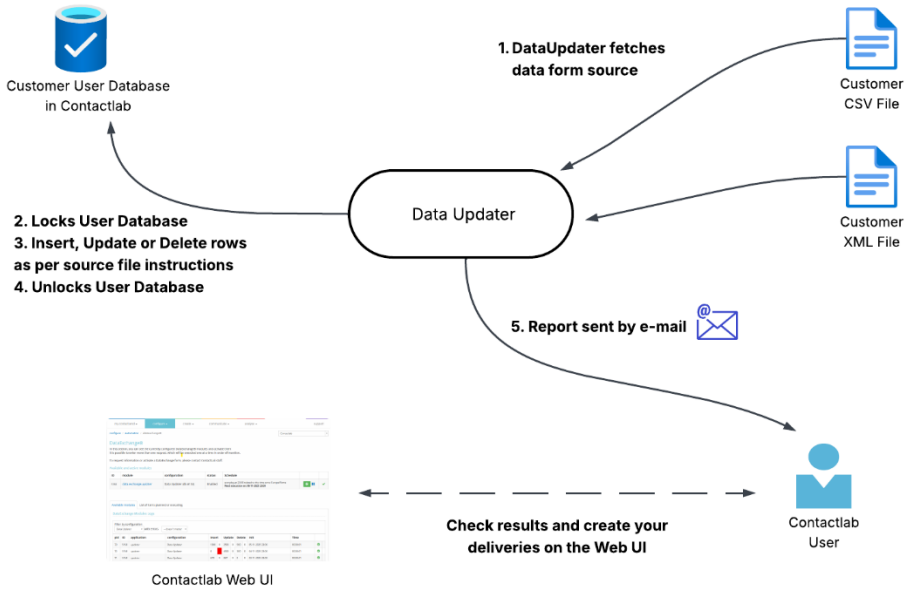
DataExchange is a name for a number of application modules dealing with automated data transfer to and from Send, possibly transforming data while it is being transferred, and the automatic setup of filters to be used in campaign definitions. All of this, without the need of accessing or operating on the web UI.
Data transfer can be bi-directional and utilize different protocols and methodologies according to each customer's unique needs and skills available in-house. Recipient databases or reporting datasets are among the types of data available through DataExchange configurations.
A set of different DataExchange modules are available to be used according to specific situations:

- **DataUpdater**: it allows to perform managed incremental updates to a specific recipient (user) database in Send, with data coming from an external source.
- **DataImporter**: it allows to load a specific recipient (user) database in Send from an external source.
- **DataCollector**: it allows to merge data from multiple text files and trigger the execution of a DataExchange module.
- **DataExporter**: it allows to export data about campaigns summary or analytical data from Send to a remote system, either as a CSV file or an XML file.
- **DataDisplacer**: it allows to export data extracted from User Database hosted in Send to a remote system, either as a CSV file or posting data to a remote database.

The above mentioned modules do not only manage recipient data or events relating to deliveries (open, clicks etc.) – they can also deal with in-depth data such as ROI events, or any other event platform can track, in order to let the customer easily setup triggered actions basing on specific conditions considering a multitude of unrelated facts coming from different systems. The following chapters offer an in-depth review of the available DataExchange modules.

2. The DataUpdater module

The DataUpdater module allows the maximum flexibility in updating user databases in Send with data coming from external sources.



2.1 Supported file format and datasource structure

The DataUpdater module needs a datasource providing data to be applied to the user database for any operation.
The datasource can be generated in various formats, but there must exist a clear relationship among the datasource's structure and the user database's field definitions in Send.
An additional and mandatory field must exist in the datasource and provide information on the action to be performed on a specific record (row) in the datasource itself. Not surprisingly, the expected name for this field is action, a text field assuming one of the following values: I (for Insert), U (for Update) and D (for Delete).

The following formats are supported:

- **CSV (comma separated value) file**: fields in the CSV file must correspond exactly both in ordering and datatypes to fields defined in the destination Send user database. The field separator, string delimiter and line-ending character can be configured according to customer needs. An additional field (action) must always be present to tell the DataUpdater module what to do with the record (row) being examined.

Example (CSV file):
Let's suppose we have a user database in Send (the destination) with the following structure and records:

"Subscribers to gardening newsletter" (please note, fictitious data):

ID	EMAIL	FIRSTNAME	LASTNAME	ADDRESS	PHONE	SEX
1	Jsmith@example.com	John	Smith	1, Rome St.	09941444	M
2	Rb@example.net	Robert	Bar	2, Bear Av.	10001433	M
3	Tinaf@example.org	Tina	Foo	3, Time Sq.	21199001	F

Let's further suppose data needs to be updated thusly: Robert Bar asked to be removed and will be deleted from the user database; Tina Foo changed its street address to "4, Past Sq."; and we have a new subscriber, Carly Fines. John Smith will remain as it is.

The following CSV-format datasource for DataUpdater will do the trick:

2;"Rb@example.net";"Robert";"Bar";"2, Bear Av.";"10001433";"M";"D"
3;"Tinaf@exaple.org";"Tina";"Foo";"4, Past Sq.";"21199001";"F";"U"
4;"Carly.f@clab1.net";"Carly";"Fines";"10, Ten St.";"67710221";"F";"I"

The final results in our user database:

“Subscribers to gardening newsletter” (please note, fictitious data):

ID	EMAIL	FIRSTNAME	LASTNAME	ADDRESS	PHONE	SEX
1	Jsmith@example.com	John	Smith	1, Rome St.	09941444	M
3	Tinaf@example.org	Tina	Foo	4, Past Sq.	21199001	F
4	Carly.f@clab1.net	Carly	Fines	10, Ten St.	67710221	F

Although field separator, string delimiter and record separator are configurable, the name of the action field cannot be changed.

- **XML file:** a single “root” node (dataroot) must include child nodes (RECORD) for every record, and every record will contain tags with name corresponding the the user databases’ field names. Values in these tags will be applied to records being imported by the DataUpdater module. Additionally, a mandatory action attribute must be specified for each RECORD node. For the XML format, it is possible to omit fields in an Update action block if their value must not necessarily be updated. The master condition field (read “2.3 Operation modes”) must always be present. Fields can also be omitted in the D action.

Example (XML file):

This is the same example for the previous section (see above); we present the XML file needed to perform exactly the same set of operations. Please note the action attribute in the opening tag for the RECORD node. In this example, ID is the master condition (read “2.3 Operation modes”):

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot>
     <RECORD ACTION=”D”>
           <ID>2</ID>
     </RECORD>
     <RECORD ACTION=”U”>
           <ID>3</ID>
           <EMAIL>Tinaf@example.org</EMAIL>
           <FIRSTNAME>Tina</FIRSTNAME>
           <LASTNAME>Foo</LASTNAME>
           <ADDRESS>4, Past Sq.</ADDRESS>
           <PHONE>21199001</PHONE>
           <SEX>F</SEX>
     </RECORD>
     <RECORD ACTION=”I”>
           <ID>4</ID>
           <EMAIL>Carly.f@clab1.net</EMAIL>
           <FIRSTNAME>Carly</FIRSTNAME>
           <LASTNAME>Fines</LASTNAME>
           <ADDRESS>10, Ten St.</ADDRESS>
           <PHONE>67710221</PHONE>
           <SEX>F</SEX>
     </RECORD>
</dataroot>
```

2.2 Location of the datasource

Once the datasource (file or remote database) is available, DataUpdater must be able to access it in order to perform its duty. The following approaches are available for this step:

- **Source file available via SFTP (suggested)**: Customer can ask Contactlab to provide an account on its own servers (strongly recommended). Contactlab will provide credentials and instructions to connect to the customer-reserved area on its servers; customer will have to setup a process to upload the file(s) in this area. FTP/FTPS protocols are not supported.
- **Source file available via HTTPS**: DataUpdater can also retrieve a named file from an https protected resource.

As an option, in addition to using an encrypted transport such as SFTP or HTTPS, it is also possible to use PGP/GPG public/private key pair encryption. Customer will have to provide its public key to Contactlab and Contactlab will provide its public key to customer.

As an option, compression and PGP/GPG encryption can be applied at the same time (in the specified order) on DataExchange files.

As an option, the module can be configured to rename the source file by automatically adding a timestamp with format YYYYMMDDHHmmss or remove it.

2.3 Operation modes

DataUpdater normally analyzes each and every record (row) in the datasource, matching the specified key (read further), and applies the action as specified in the properly named field in the source.

When updating (U) records being updated must already be present in the destination table. If they are not present, DataUpdater can either insert them as new or ignore them. These two different behaviors can be configured. With the auto-insert configuration, there is no need to use the 'I' operation, although using it might help troubleshooting as it clearly declares the intended operation and could help identify discrepancies between the master source and the user database in Send.

Identifying records is an important part of the DataUpdater operations. It can be configured to use one or two fields as "keys"; we name this specification *Master Condition*.

Example (Master Condition):

Suppose we have the following user database as a destination. It is very similar to our "gardening" example, but we now have the ability to manage multiple newsletters using a campaign_type field. This means the same email might be present more than once in this table.

"Subscribers to all newsletters" (please note, fictitious data)

EMAIL	CAMPAIGN_TYPE	FIRSTNAME	LASTNAME	ADDRESS	PHONE	SEX
Jsmith@example.com	Gardening	John	Smith	1, Rome St.	09941444	M
Tinaf@example.org	Motors	Tina	Foo	3, Time Sq.	21199001	F
Jsmith@example.com	Motors	John	Smith	1, Rome St.	09941444	M

John Smith subscribed to both the Garderning and Motors-themed newsletters. At some point in time, he updates his profile on the customers' website, updating is subscription data for Gardening (the site is organized by theme). Also, a new subscription arrives, and Tina Foo updates her street address. The following datasource sample includes the changes we just described:

”Jsmith@example.com”;”Gardening”;”John”;”Smith”;”2, New St.”;09941444;”M”;”U”
”Tinaf@example.org”;”Motors”;”Tina”;”Foo”;”4, Past Sq.”;”21199001”;”F”;”U”
”Carly.f@clab1.net”;”Gardening”;”Carly”;”Fines”;”10, Ten St.”;”67710221”;”F”;”I”

The *Master Condition* tells the DataUpdater module which key to use to identify records to be modified (updated). If the *Master Condition* in this example includes the EMAIL field only, both records for Jsmith@example.com will be updated, producing the following result:

“Subscribers to all newsletters” (please note, fictitious data)

EMAIL	CAMPAIGN_TYPE	FIRSTNAME	LASTNAME	ADDRESS	PHONE	SEX
Jsmith@example.com	Gardening	John	Smith	2, New St.	09941444	M
Tinaf@example.org	Motors	Tina	Foo	4, Past Sq.	21199001	F
Carly.f@clab1.net	Gardening	Carly	Fines	10, Ten St.	67710221	F
Jsmith@example.com	Gardening	John	Smith	2, New St.	09941444	M

If, instead, the Master Condition includes the couple of fields [EMAIL, CAMPAIGN_TYPE], only one of two records for John Smith will be updated. The following will be the final result:

“Subscribers to all newsletters” (please note, fictitious data)

EMAIL	CAMPAIGN_TYPE	FIRSTNAME	LASTNAME	ADDRESS	PHONE	SEX
Jsmith@example.com	Gardening	John	Smith	2, New St.	09941444	M
Tinaf@example.org	Motors	Tina	Foo	4, Past Sq.	21199001	F
Carly.f@clab1.net	Gardening	Carly	Fines	10, Ten St.	67710221	F
Jsmith@example.com	Motors	John	Smith	1, Rome St.	09941444	M

Please notice the also the ADDRESS field for [‘Jsmith@example.com’, Motors] remained ‘1, Rome St.’ (instead of ‘2, New St.’ since the update only identifies [‘Jsmith@example.com’, Gardening] as the record to be updated.

2.4 Additional tools and configuration options

DataUpdater can either remove/clean the datasource (in case of a remote table, delete all records in the table) after the import operation has completed, or leave it untouched; this can be configured as needed.

ETL (extract, transform and load) options are available; in case a specific ETL is required which is not available as a configurable item, a customization can be developed according to customer requirements. An ETL are routinely utilized to convert and normalize data coming from different datasources and destined to a single Send user database.

Post-operation actions (post-actions) can also be specified as a form of ETL to perform highly customized maintenance operations (see “10 DataExchange Post-Actions”).

2.5 Execution and reporting options

A DataUpdater operation can be launched manually using the web UI (“Configure > Automation dataexchange”) or scheduled to run periodically. In the latter case, the configuration must be initially setup (activated) by Contactlab; once periodic execution is activated, customer will be able to modify it in the web UI.

It is possible to request that executions are retried for a certain number of times before giving up and considering it an error.

It is possible to request that an error does not lock the User Database. Please consider the implications (your subsequent deliveries will operate on a possibly out of data database).

After the operation is concluded, an automated summary report on results will be sent via e-mail (optionally). Customers can autonomously configure one recipient for these summary reports. It is also possible to view the summary report’s contents in the web UI, in the same section.

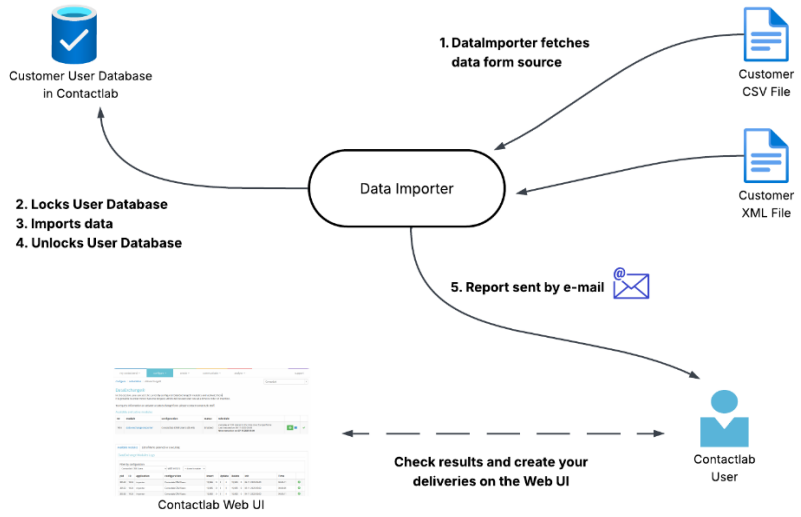
The same recipient will receive alerts on errors or warning if an operation was not completely successful; diagnostic log files can be download from the Web UI - please refer to section “10. Diagnostics messages” for more information on diagnostic messages and their meaning.

The web UI user account under which the DataUpdater is configured (the DataUpdater “owner”) will also be able to view in real-time progress and notifications on DataUpdater operations. This is only available for the owner of the configuration. Every DataUpdater configuration has one and only one owner; it can be specified when requesting activation of this module.

3. The DataImporter module

The DataImporter module allows loading, in append or refresh mode, a recipient/user database in Send with data generated from external sources.

IMPORTANT NOTICE. The DataUpdater module should always be preferred over DataImporter. DataImporter will not preserve primary keys when used in refresh mode and will generate many duplicates of the same records when used in append mode. Please consider using DataUpdater instead of DataImporter whenever possible.



3.1 Supported file formats and datasource structure

The DataImporter module needs a datasource providing data to be applied to the user database for any operation. The datasource can be generated in various formats, but there must exist a clear relationship among the datasource's structure and the user database's field definitions in Contactlab. The following formats are supported:

- **CSV (comma separated value) file**: fields in the CSV file must correspond exactly both in ordering and datatypes to fields defined in the destination Send user database. The field separator, string delimiter and line-ending character can be configured according to customer needs.

- **XML file**: a single "root" node (dataroot) must include child nodes (RECORD) for every record, and every record will contain tags with name corresponding the user databases' field names. Values in these tags will be applied to records being imported by the DataImporter module.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<dataroot>
     <RECORD>
           <EMAIL>Foo.Bar@example.com</EMAIL>
           <FIRSTNAME>Foo</FIRSTNAME>
           <LASTNAME>Bar</LASTNAME>
           <ADDRESS>via roma 123</ADDRESS>
           <PHONE>3291234567</PHONE>
           <SEX>m</SEX>
     </RECORD>
     <RECORD>
….
…
</RECORD>
…
…

</dataroot>
```

In this example, the destination user database will be composed of fields EMAIL, FIRSTNAME, LASTNAME, ADDRESS, PHONE, SEX; they will be populated with tag values taken from each record found in the XML file.

3.2 Location of the datasource

Once the datasource is available, DataImporter must be able to access it in order to perform its duty. The following approaches are available for this step:

- **Source file available via SFTP (suggested)**: Customer can ask Contactlab to provide an account on its own servers (strongly recommended). Contactlab will provide credentials and instructions to connect to the customer-reserved area on its servers; customer will have to setup a process to upload the file(s) in this area. FTP/FTPS protocols are not supported.

- **Source file available via HTTPS:** DataImporter can also retrieve a named file from an http/https protected resource.

As an option, in addition to using an encrypted transport such as SFTP or HTTPS, it is also possible to use PGP/GPG public/private key pair encryption. Customer will have to provide its public key to Contactlab and Contactlab will provide its public key to customer.

As an option, compression and PGP/GPG encryption can be applied at the same time (in the specified order) on DataExchange files.

As an option, the module can be configured to rename the source file by automatically adding a timestamp with format YYYYMMDDHHmmss or remove it.

3.3 Operation modes

DataImporter is usually setup for a full refresh operation mode: every time a new datasource is available, it will first completely empty the destination user database, and then insert all records from the datasource.

It is also possible to configure the DataImporter in append mode: in this configuration, the module will not wipe out all contents of the destination user database, but simply add records found in the datasource to previous content. In this configuration, it is also possible to specify a “deletion filter” so that only records matching the filter will be deleted prior to appending the content of the datasource.

3.4 Limitations

If DataImporter module is setup for a full refresh and the related User Database hosted in Send doesn’t have a customer managed Primary Key, the following problems may occur:

- Impossibility to aggregate events by user: since a unique id is not maintained, it’s impossible to recognize user events (such as bounces, opens or views) of different deliveries.

- Newsletter online version may not work correctly and dynamic fields may not be available.

For these reasons, we strongly recommend to activate a DataUpdater instead of a DataImporter module.

3.5 Additional tools and configuration options

DataImporter can either remove/clean the datasource after the import operation has completed, or leave it untouched; this can be configured as needed.

ETL (extract, transform and load) options are available; in case a specific ETL is required which is not available as a configurable item, a customization can be developed according to customer requirements. An ETL are routinely utilized to convert and normalize data coming from different datasources and destined to a single Send user database.

Post-operation actions (post-actions) can also be specified as a form of ETL to perform highly customized maintenance operations (see “9. DataExchange Post-Actions”)

3.6 Execution and reporting options

A DataImporter operation can be launched manually using the web UI (“Configure > Automation > dataexchange”) or scheduled to run periodically. In the latter case, the configuration must be initially setup (activated) by Contactlab; once periodic execution is activated, customer will be able to modify it in the web UI.

After the operation is concluded, an automated summary report on results will be sent via e-mail (optionally). Customers can autonomously configure one recipient for these summary reports. It is also possible to view the summary report’s contents in the web UI, in the same section.

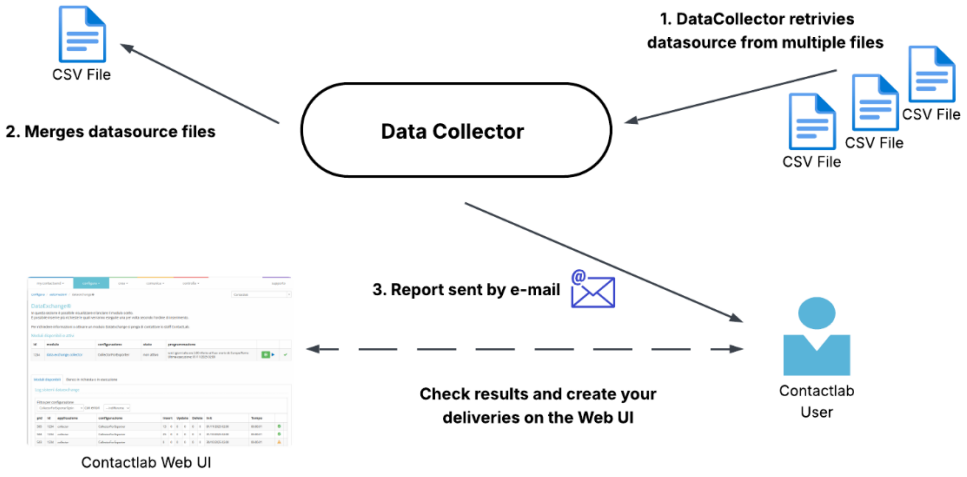
The same recipient will receive alerts on errors or warning if an operation was not completely successful; diagnostic log files can be download from the Web UI - please refer to section “11. Diagnostics messages” for more information on diagnostic messages and their meaning.

The web UI user account under which the DataImporter is configured (the DataImporter “owner”) will also be able to view in real-time progress and notifications on DataImporter operations. This is only available for the owner of the configuration. Every DataImporter configuration has one and only one owner; it can be specified when requesting activation of this module.

4. The DataCollector module

The DataCollector module allows merging data from multiple text files and then trigger the execution of a DataExchange module.

This is required when you need to feed a DataUpdater or a DataImporter with datasource available on multiple files.



4.1 Supported file formats and datasource structure

The DataCollector module needs one or more CSV (comma separated value) files as datasource.

The module can be configured to process and merge data provided in a specific order, based on files attributes or custom rules.

The output file must be a valid datasource for the DataExchange module triggered after each DataCollector execution (See related documentation for details).

As an option, CSV first line of specific datasource files can be removed from the output.

4.2 Access and location of the datasource

DataCollector searches for all files matching a specific regular expression inside a given directory and use them as datasource.

Once the datasource is available, DataCollector must be able to access it via SFTP. Customer can ask Contactlab to provide an account on its own servers (strongly recommended). Contactlab will provide credentials and instructions to connect to the customer-reserved area on its servers; customer will have to setup a process to upload the file(s) in this area. FTP/FTPS protocols are not supported.

As an option, the module can be configured to rename source files by automatically adding a timestamp with format YYYYMMDDHHmmss or remove them.

As an option, the module can be configured to rename output file by automatically adding a timestamp with format YYYYMMDDHHmmss.

4.3 Additional tools and configuration options

DataCollector can either remove/clean the datasource after the merge operation has completed or leave it untouched; this can be configured as needed.

After its execution, DataCollector can trigger the execution of another DataExchange module.

4.4 Execution and reporting options

A DataCollector operation can be launched manually using the web UI ("Configure > Automation dataexchange") or scheduled to run periodically. In the latter case, the configuration must be initially setup (activated) by Contactlab; once periodic execution is activated, customer will be able to modify it in the web UI.

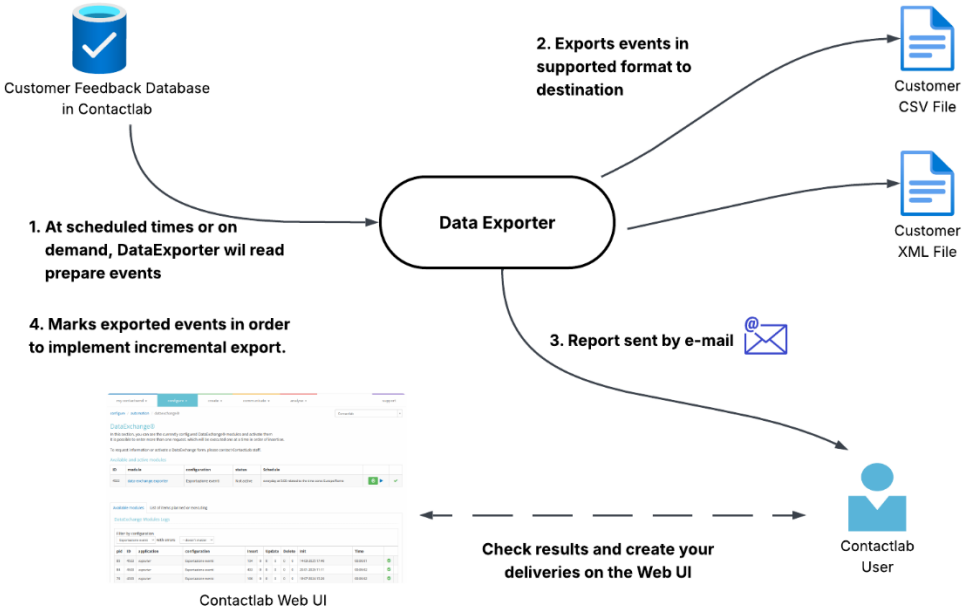
After the operation is concluded, an automated summary report on results will be sent via e-mail (optionally). Customers can autonomously configure one recipient for these summary reports. It is also possible to view the summary report's contents in the web UI, in the same section.

The same recipient will receive alerts on errors or warning if an operation was not completely successful; diagnostic log files can be download from the Web UI - please refer to section "10. Diagnostics messages" for more information on diagnostic messages and their meaning.

The web UI user account under which the DataCollector is configured (the DataCollector "owner") will also be able to view in real-time progress and notifications on DataCollector operations. This is only available for the owner of the configuration. Every DataCollector configuration has one and only one owner; it can be specified when requesting activation of this module.

5. The DataExporter module

While DataUpdater, DataImporter deal with transferring data *to* Contactlab from external sources, the DataExporter allows to export data *from* Send to a customer repository.



With DataExporter, Contactlab customers gain access to data about two important indicators for campaigns: campaign-level aggregated feedback statistics, and recipient-level details for feedback events. Detailed info on available feedback indicators are available in this section.

Recipient-level details and campaign-level aggregations are generated continuously, with an estimated maximum delay of 30 minutes between event registration and data availability for exporting.

5.1 Supported file formats

DataExporter can generate datasets in the following formats:

- **CSV file**: a standard comma-separated-value with a header line (the first line in the files will contain the names of the fields). Field separator, string delimiter and line terminator characters can be configured according to customer needs. Also, the generated file name can be specified by the customer.

- **XML file**: files will include a single “root” node called ROOT with multiple child nodes (RECORDELEMENT) for every record.

5.2 Location of the destination

Contactlab can provide DataExporter results (datasets) delivering them to **SFTP**:

- Customer can ask Contactlab to provide such an account on its own servers (strongly recommended). Contactlab will provide credentials and instructions to connect to the customer-reserved area on its servers; customer will have to setup a process to upload the file(s) in this area. FTP/FTPS protocols are not supported.

As an option, in addition to using an encrypted transport such as SFTP or HTTPS, it is also possible to use PGP/GPG public/private key pair encryption. Customer will have to provide its public key to Contactlab and Contactlab will provide its public key to customer.

As an option, compression and PGP/GPG encryption can be applied at the same time (in the specified order).

5.3 Operations mode and available information

The preparation of the feedback dataset can be “full” or “incremental”. If “full”, *all* existing data in Send will be exported every time; if “incremental”, DataExporter keeps track of the last export operations timestamp and exports all data generated or modified *after* the last operation.

The “full” setup is available but only suggested for small datasets (small number of total messages sent in a typical monthly period) and/or to temporarily activate a feedback export while the “incremental” setup is fine-tuned. Although no specific limit is imposed on the amount of data, if the customer expects to use the “full” export for more than six months, special dispositions must be agreed with Contactlab at setup time.

There are three different datasets available for DataExporter. Customer can choose which are interesting to his/her needs. Also, each dataset has different fields available. Once again, customer can choose which fields are to be exported for each dataset.

Delivery dataset: aggregated campaign (delivery)-level key performance indicator. The following table summarizes available indicators (fields in the dataset). It is also possible to request a different ordering of fields at DataExporter configuration time.

FIELD	DESCRIPTION	TYPE	FORMAT
delivery_id	Unique id for delivery	Numeric	
newsletter	Name of newsletter/group used in delivery	Alphanumeric	Variable-length string (255 max)
subject	Delivery subject	Alphanumeric	Variable-length string (255 max)
start_date	Date and time of start of delivery	ISO-format timestamp	yyyy-MM-dd hh:mm:ss
Recipients	Total number of recipients	Numeric	
description	Qualifier/filter descriptive name	Alphanumeric	Variable-length string (255 max)
newsletter_id	group/newsletter id		
userdb_id	user database unique id	Numeric	
charset	Id of charset used	Alphanumeric	Variable-length string (255 max)
role	“role” of delivery in a multi-segment delivery (e.g. spllt delivery, A/B testing)	Numeric	0: non-segmented 1: master of segmented 2: master of A/B test 101-199: segment 200: “winner” segment of A/B test
parent_id	Id of “master” in multi-segment delivery	Numeric	
medium	Delivery medium/channel type	Numeric	0: email 1: sms 2: fax
title	Delivery title in Send	Alphanumeric	Variable-length string (255 max)
category_id	Id of “master” in multi-segment delivery	Numeric	
qualifier_id	Id if qualifier/filter	Numeric	
end_date	Date and time of start of delivery	ISO-format timestamp	yyyy-MM-dd hh:mm:ss
mail_from	RFC5322 From (“Friendly From”)	Alphanumeric	Variable-length string (255)
reply_to	RFC 5322 Reply-To header value	Alphanumeric	Variable-length string (255)
schedule_date	Date and time of scheduling (if any)	ISO-format timestamp, or empty	yyyy-MM-dd hh:mm:ss or
remove_dupli	Flag: remove duplicate addresses from broadcast queue	Numeric	0 or 1
is_test	Flag: “Test” delivery	Numeric	0 or 1
track_urls	Flag: click tracking enabled	Numeric	0 or 1
read_stats	Flag: open/view tracking enabled	Numeric	0 or 1
Communication category_id	Communication Category Id	Numeric	1 or 6

The *role* and *parent_id* fields are meaningful for multi-segment deliveries only, e.g. “split deliveries” or “A/B testing” deliveries. These deliveries are managed by setting up a “master” delivery acting as a container/parent for all segments. When a record is referring to a segment of a multi-segment delivery, the *role* field will have a value between 101 and 199 and *parent_id* will contain the delivery_id of the master delivery. When a record is referring to the master delivery, the role field will be valued 1. For non-segmented deliveries, the *role* field will be set to 0 (zero).

The *category_id* field will specify the Category of the delivery as assigned by the user in the Web UI. If you are interested in this feature, please ask our Customer Care for the current list of allowed categories (as seen in the web UI) and their numerical ids.

The *communication_category_id* reports the "Communication Category". Currently supported communication categories are 1 (Marketing/Bulk) and 6 (Transactional).

Delivery_info dataset: specifically for delivery notes/labels (key-value pairs customer can attach to any delivery, usually utilized to classify deliveries to enable integration in external customer applications).

FIELD	DESCRIPTION	TYPE	FORMAT
delivery_info_id	Unique id for this record	Numeric	
delivery_id	Unique id for delivery	Numeric	
label	Label/note descriptive text (key)	Alphanumeric	Variable-length string (255 max)
value	Label/note text value (value)	Alphanumeric	Variable-length string (255 max)
is_system	Flag: this is a system key	Alphanumeric	Variable-length string (255 max)

Event_info daset: detailed information on all recipients' actions (events) tracked for a delivery.

FIELD	DESCRIPTION	TYPE	FORMAT
delivery_id	Unique delivery (campaign) id	Numeric	
event_id	Unique delivery id	Numeric	
event_code	Event code	Character	Single Character
event_details	Event detail	Alphanumeric	Variable-length string (255)
event_subdetails	Event subdetail	Alphanumeric	Variable-length string (255 max)
category_id	Event category id	Numeric	
category_desc	Event category description	Alphanumeric	Variable-length string (255)
event_date	Event date and time	ISO-format timestamp	yyyy-MM-dd hh:mm:ss
user_pk	Primary key of recipient in user DB	Numeric	
ip_address	IPv4 or IPv6 address of host	ip-address	Variable-length string (255)
user_agent	User agent signature	Alphanumeric	Variable-length string (255)

The event_code can assume one of the following values (please notice the character code is **case sensitive** – event code "s" has a different meaning than event code "S"):

- A: an abuse complaint was filled by recipient using the form referenced in the X-Abuse header.
- B: "bounce" event (a bounce was received when trying to deliver the message).
- C: "click" event (a tracked link was clicked).
- D: an abuse complaint was filled by recipient in a non-structured way (e.g. mail to abuse contact, general-purpose abuse complaint form…).
- E: "error" event (email address was invalid or other causes).
- F: feedback loop, recipient filled a complaint via a feedback loop feed.
- G: "consent granted" event (tracking consent was set to true), available when consent is managed through Contactlab and the __clt field is present.
- I (uppercase i): Delivery was interrupted, so this message was not sent.
- J: temporary blacklist match: message was NOT sent because recipient was found in temporary blacklist.
- K: permanent blacklist match: message was NOT sent because recipient was found in a permanent blacklist.
- L: list-unsubscribe, recipient asked for unsubscription via List-Unsubscribe header automation.
- m: profile update, available if events of this type are configured to be tracked.
- o: optout (unsubscribe), available if events of this type are configured to be tracked (requires Contactlab's integrated Subscription System).
- s: optin (subscribe), available if events of this type are configured to be tracked (requires Contactlab's integrated Subscription System).
- R: "consent revoked" event (tracking consent was set to false), available when consent is managed through Contactlab and the __clt field is present.
- S: "sent" event for bilk email message.
- T: "sent" event for a triggered, sendImmediate email message or for a Contactlab Smart Relay managed message.
- V: "view" or "open" event (the open/view tracker marker URL was seen by Contactlab).

- X: recipient entered the blacklist/suppression list because of a non-delivery-related event2.
- Y: recipient entered the temporary blacklist/suppression list as a result of an event.
- Z: recipient entered the permanent blacklist/suppression list. This is the fourth consecutive bounce or any "bad destination mailbox address" bounce. It replaces the previous 'X' event.

Event_id is the primary key for events: it is unique for events for a given customer account, even across different user databases for the same customer. Together with *event_date, event_code* and *event_details*, it will always have a value.

Ip address and *user agent* instead are filled when available (when the event is caused by a user action from an Internet-connected device). The *event_details, event_subdetails, category_id* and *category_desc* fields can assume different meanings according to the event code.

The DataExporter standard configuration will export *all* available information (event records for all event codes). As an optional configuration, events can be filtered by Send before the exporter stream is generated in order to include events the customer is interested in.

Custom filters on events to be exported can be requested to your Account/Sales manager and will be estimated as custom development work.

Event details will show some differences according to the medium of the campaign. Let's examine all available choices to simplify matters.

5.3.1 Event info details for email messages

The following table summarizes the combinations:

EVENT CODE	EVENT DETAILS	EVENT SUBDETAILS	CATEGORY ID	CATEGORY DESC	IP ADDRESS	USER AGENT	RECIPIENT (*)
A	Reason – free text, entered by user	Recipient Email domain	—	—	Origin IP address of http reg.	User-agent header of http reg.	√
B	Short description of bounce	Bounce class: hard (H), soft (S)	Bounce category code	—	—	—	√
C	Tracked link destination URL	Link descry as set up in Send UI	Link category id	Link category name	Origin IP address of http reg.	User-agent header of http reg.	√
D	Complainant email address	—	—	Notes	—	—	√
E	Recipient email address	—	—	—	—	—	√
F	—	Recipient email domain	—	—	—	—	√
G	Recipient email address	User DB id	—	—	Origin IP address of http reg.	User-agent header of http reg.	√
I	Recipient email address	Recipient email domain	—	—	—	—	√
J	Recipient email address	—	—	—	—	—	√
K	Recipient email address	Blacklist id	Blacklist category	—	—	—	√
L	—	Recipient email domain	—	—	Origin IP address of http reg.	User-agent header of http reg.	√
m	Recipient email address	—	—	—	Origin IP address of http reg.	User-agent header of http reg.	√
o	Recipient email address	Subscription form id	—	—	Origin IP address of http reg.	User-agent header of http reg.	√
R	Recipient email address	User DB id	—	—	Origin IP address of http reg.	User-agent header of http reg.	√
s	Recipient email address	—	—	—	Origin IP address of http reg.	User-agent header of http reg.	√

S	Recipient email address	Recipient email domain	—	—	—	—	√
T	Recipient email address	Recipient email domain	—1	—2	—	—	√
V	Webmail type	—	—	Device identified on data	Origin IP address on http reg.	User-agent header of http reg.	√
X	Recipient email address	Recipient email domain	Reason code	—	—	—	√
Y	Recipient email address	Number of consecutive bounces	Reason code	Blacklisted until timestamp	—	—	√
Z	Recipient email address	Number of consecutive bounces	Reason code	—	—	—	√

(*) Recipient field. This field is available as an optional configuration and will always contain the Recipient email address. This enhancement was implemented to simplify exported events ingestion at customer end. In order to guarantee backward compatibility, the same information will still be inserted in the "Event Details" field as reported in the above table, so that existing ingestion procedure will still work as expected.

5.3.2 Event info details for SMS messages

SMS messages support the following event codes:

- B: "bounce" event - a negative Delivery Report (DR) notification was received when trying to deliver the message. We are currently reporting the verbatim description and code from the operator, which do not have a standard classification.
- C: "click" event (a tracked link was clicked).
- E: "error" event (phone number was invalid or other causes).
- I (uppercase i): Delivery was interrupted, so this message was not sent.
- S: "sent" event for bulk SMS message.
- T: "sent" event for a triggered, sendImmediate SMS message or for a Contactlab Smart Relay managed message.
- V: for SMS messages, this represents a positive Delivery Report (DR) notification received for this message.

Please be aware that currently we do not support blacklist functionality for SMS message. This means the J, X, Y, and Z event codes will never actually be found in a DataExporter feed for SMS messages; we however suggest to consider them in your import procedures as a no-op to prepare for future upgrades.

The following table summarizes the combinations (including the I, J, X and Y events):

EVENT CODE	EVENT DETAILS	EVENT SUBDETAILS	CATEGORY ID	CATEGORY DESC	IP ADDRESS	USER AGENT	RECIPIENT
B	Short description of bounce	Bounce class: hard (H), soft (S)	Bounce category code	—	—	—	√
C	Tracked link destination URL	Link descry as set up in Send UI	Link category id	Link category name	Origin IP address of http reg.	User-agent header of http reg.	√
E	Recipient MSISDN	—	—	—	—	—	√
I	Recipient MSISDN	—	—	—	—	—	√
S	Recipient MSISDN	—	—	—	—	—	√
T	Recipient MSISDN	—	—3	—4	—	—	√
V	—	—	—	—	—	—	√

1 For SmartRelay send events, this field can be populated by customer using custom header in the message as an optional feature.
2 For SmartRelay send events, this field can be populated by customer using custom header in the message as an optional feature.
3 For SmartRelay send events, this field can be populated by customer using custom header in the message as an optional feature.
4 For SmartRelay send events, this field can be populated by customer using custom header in the message as an optional feature.

5.3.3 Device Identification Data for “view” (V) events

The “Category Desc” field for event type “V” (view) is populated with device identification data from the ScientiaMobile WURFL device detection database.[5]

To ease automated parsing of these data, subfields for the device identification are concatenated with character “~” (tilde) as a field separator.

Subfields are provided in the following order:

- Send internal brand/vendor id (an integer number)
- Brand/vendor “human readable” name
- Send internal model id (an integer number)
- Device Model “human readable” name
- Device Model “marketing” name
- Mobile device flag (1 if device is mobile, 0 otherwise)
- Tablet device flag (1 if device is a tablet, 0 otherwise)
- Operating System name
- Operating System version
- Browser name
- Browser version

When a subfield value is not available it will be left empty so you can use a positional index to access subfields after separating them.

A few real-world samples:
6~Apple~5091~iPad~~1~1~iPhone OS~6.0~Safari~
59~Microsoft~2~Internet Explorer~~0~0~~~MSIE~7.0
16~Google~6573~Chrome~~0~0~Desktop~0~~23.0
14~Samsung~4517~GT-I9000~Galaxy S~1~0~Android~2.3~Android Webkit~

The following data is provided when the device was not correctly identified:
1~generic web browser~1~~~0~0~~~~

5.3.4 Blacklist identification data in ‘blacklist match’ (K) events

The ‘blacklist match’ (K) event, which records all messages not sent because the recipient was in a supported permanent blacklist/suppression list at the time the campaign was processed, will provide additional info on *which* blacklist/suppression list matched. Field ‘Category id’ will have a value according to the following:

- 101 if there was a match in Send per-customer blacklist/suppression list.
- 103 if there was a match in a custom suppression list13.

When ‘Category id’ is 103, field ‘Event subdetails’ will have a value according to the following:

- No value.
- Unique id of the custom suppression list.

5 Contactlab is a licensed an adopter of ScientiaMobile WURFL products. Contactlab license covers usage of WURFL-derived data inside our platform, but you can also use this information to generate custom reporting outside of Contactlab platform, as long as you are an active customer of Contactlab.

5.4 XML file format specifications

5.4.1 Delivery Dataset

```
<?xml version="1.0" encoding="UTF-8"?>
<ROOT>
     <RECORDELEMENT>
           <DELIVERY_ID>3034</DELIVERY_ID>
           <NEWSLETTER>A sample mail</NEWSLETTER>
           <SUBJECT>
                 <![CDATA[A sample mail]]>
           </SUBJECT>
           <START_DATE/>
           <RECIPIENTS>14</RECIPIENTS>
           <DESCRIPTION>
                 <![CDATA[Test join]]>
           </DESCRIPTION>
           <CATEGORY_ID>0</CATEGORY_ID>
     </RECORDELEMENT>
     <RECORDELEMENT>
…
</RECORDELEMENT>
</ROOT>
```

5.4.2 Delivery Info Dataset

```
<?xml version="1.0" encoding="UTF-8"?>
<ROOT>
     <RECORDELEMENT>
           <DELIVERY_INFO_ID>1</DELIVERY_INFO_ID>
           <DELIVERY_ID>26</DELIVERY_ID>
           <LABEL>label1</LABEL>
           <VALUE>test label</VALUE>
           <IS_SYSTEM>false</IS_SYSTEM>
     </RECORDELEMENT>
</ROOT>
```

5.4.3 Event Info Dataset

```
<ROOT>
     <RECORDELEMENT>
           <DELIVERY_ID>1357</DELIVERY_ID>
           <NEWSLETTER>
                 <![CDATA[Newsletter Email Group]]>
           </NEWSLETTER>
           <SUBJECT>
                 <![CDATA[New offers for you]]>
           </SUBJECT>
           <START_DATE>2025-11-19 12:42:20.0</START_DATE>
           <RECIPIENTS>205891</RECIPIENTS>
           <DESCRIPTION>
                 <![CDATA[Qualifier or Filter Description]]>
           </DESCRIPTION>
           <NEWSLETTER_ID>17</NEWSLETTER_ID>
           <USERDB_ID>17</USERDB_ID>
           <ROLE>0</ROLE>
           <PARENT_ID></PARENT_ID>
           <MEDIUM>0</MEDIUM>
           <TITLE></TITLE>
           <CATEGORY_ID>0</CATEGORY_ID>
           <QUALIFIER_ID>10</QUALIFIER_ID>
           <END_DATE>2025-11-19 12:48:42</END_DATE>
           <MAIL_FROM>
                 <![CDATA[New offers <offers@Contactlab.com>]]>
           </MAIL_FROM>
           <REPLY_TO>
                 <![CDATA[offers@contactlab.com]]>
```

```
            </REPLY_TO>
            <SCHEDULE_DATE></SCHEDULE_DATE>
            <REMOVE_DUPLICATES>1</REMOVE_DUPLICATES>
            <IS_TEST>0</IS_TEST>
            <TRACK_URLS>1</TRACK_URLS>
            <READ_STATS>1</READ_STATS>
            <COMMUNICATION_CATEGORY_ID></COMMUNICATION_CATEGORY_ID>
    </RECORDELEMENT>
    <RECORDELEMENT>
    …
    </RECORDELEMENT>
```


5.5 Execution and reporting options

A DataExporter operation can be launched manually using the web UI ("Configure > Automation > dataexchange") or scheduled to run periodically. In the latter case, the configuration must be initially setup (activated) by Contactlab; once periodic execution is activated, customer will be able to modify it in the web UI, but it is not possible to activate periodic execution from the web UI.

After the operation is concluded, an automated summary report on results will be sent via e-mail (optionally). Customers can autonomously configure one recipient for these summary reports. It is also possible to view the summary report's contents in the web UI, in the same section.

The same recipient will receive alerts on errors or warning if an operation was not completely successful; diagnostic log files can be download from the Web UI - please refer to section "10. Diagnostics messages" for more information on diagnostic messages and their meaning.

The web UI user account under which the DataExporter is configured (the DataExporter "owner") will also be able to view in real-time progress and notifications on DataExporter operations. This is only available for the owner of the configuration. Every DataExporter configuration has one and only one owner; it can be specified when requesting activation of this module.

5.6 Data subject to variation over time

Some of the data exported by the DataExporter can possibly be modified by user actions after it has been exported. Obviously, delivery statistics can change if recipients act after a period of time. This should be considered carefully when justifying decisions basing on data aggregated too "early".

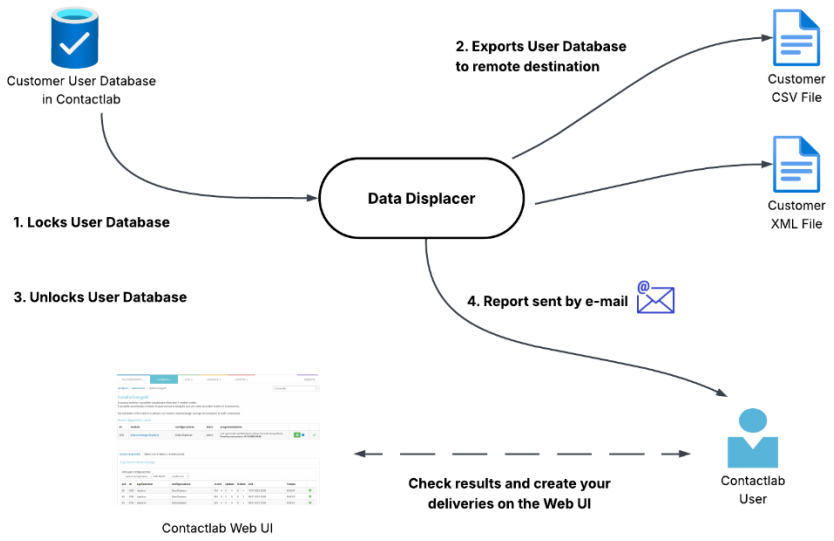
Also, customer can perform actions in Send which might change data used to identify some of the values in the available datasets (lookup fields).

For instance, if a customer deletes a delivery in Send, all related event and campaign-level info will not be available anymore. Likewise, modifying link categories and their descriptions will immediately be reflected in subsequent exports, even though they might refer to events (e.g. clicks) happened before the modification took place. In case of a full export, previously classified events might change their classification and assume new ones.

Please consult our CustomerCare or your Account whenever you have doubts with regard a DataExporter configuration, and/or you have unusual needs for data historical consistency.

6. The DataDisplacer module

The DataDisplacer module allows exporting a recipient / user database in Send to a remote system.



6.1 Supported file formats and destination structure

The DataDisplacer module allows extracting the contents of a User Database in Send. This might be needed to re-sync the source if some data (e.g. unsubscribe markers) are updated only in Send and needs to be transferred to customer. Or, a custom form might be implemented to modify user profile info in Send, so customer would like to have its master database updated regularly with updated profile info. The name DataDisplacer has been chosen to avoid confusion with the DataExporter, but by all means, it is an export operation indeed. DataExporter exports events data, while DataDisplacer exports User Databases.

The following destination formats are supported:

- **CSV (comma separated value) file**: fields in the CSV file will correspond both in ordering and datatypes to fields defined in the destination Send user database. The field separator, string delimiter and line-ending character can be configured according to customer needs.

- **XML file**: a single “root” node (dataroot) must have child nodes (RECORD) for every record, and every record will contain tags with name corresponding the user databases’ field names. Value in these tags will be applied to records being imported by the DataImporter module.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ROOT>
     <RECORDELEMENT>
           <EMAIL>someone.outthere@Contactlab.com</EMAIL>
           <FIRSTNAME/>
           <LASTNAME/>
           <__cls1>0<__ cls1>
     </RECORDELEMENT>
     <RECORDELEMENT>
           <EMAIL>same.chap@tomatointeractive.it</EMAIL>
           <FIRSTNAME/>
           <LASTNAME/>
           <__cls1>1</__cls1>
     </RECORDELEMENT>
     <RECORDELEMENT>
           <EMAIL>Xt1eZebW@Contactlab.com</EMAIL>
           <FIRSTNAME>SlZDtOSR</FIRSTNAME>
           <LASTNAME>yjkoRMN4ag</LASTNAME>
           <__cls1>-1</__cls1>
     </RECORDELEMENT>
     …
     </ROOT>
```

In this example, the User Database is composed of fields EMAIL, FIRSTNAME, and LASTNAME; they will be populated with tag values taken from each record found in the XML file. __cls1 is a special field managed by Contactlab's integrated Subscription System.

6.2 Location of the destination

Send can deliver DataDisplacer results (exported datasets) delivering them to **SFTP**:

- Customer can ask Contactlab to provide such an account on its own servers (strongly recommended). Contactlab will provide credentials and instructions to connect to the customer-reserved area on its servers; customer will have to setup a process to upload the file(s) in this area. FTP/FTPS protocols are not supported.

As an option, in addition to using an encrypted transport such as SFTP or HTTPS, it is also possible to use PGP/GPG public/private key pair encryption. Customer will have to provide its public key to Contactlab and Contactlab will provide its public key to customer.

As an option, compression and PGP/GPG encryption can be applied at the same time (in the specified order).

6.3 Additional tools and configuration options

ETL (extract, transform and load) options are available; in case a specific ETL is required which is not available as a configurable item, a customization can be developed according to customer requirements. Post-operation actions can also be specified to perform highly customized maintenance operations.

6.4 Execution and reporting options

A DataDisplacer operation can be launched manually using the web UI ("Configure > Automation > dataexchange") or scheduled to run periodically. In the latter case, the configuration must be initially setup (activated) by Contactlab; once periodic execution is activated, customer will be able to modify it in the web UI.

After the operation is concluded, an automated summary report on results will be sent via e-mail (optionally). Customers can autonomously configure one recipient for these summary reports. It is also possible to view the summary report's contents in the web UI, in the same section.

The same recipient will receive alerts on errors or warning if an operation was not completely successful; diagnostic log files can be download from the Web UI - please refer to section "10. Diagnostics messages" for more information on diagnostic messages and their meaning.

The web UI user account under which the DataDisplacer is configured (the DataDisplacer "owner") will also be able to view in real-time progress and notifications on operations. This is only available for the owner of the configuration. Every DataDisplacer configuration has one and only one owner; it can be specified when requesting activation of this module.

7. Managing DataExchange modules in Contactlab UI

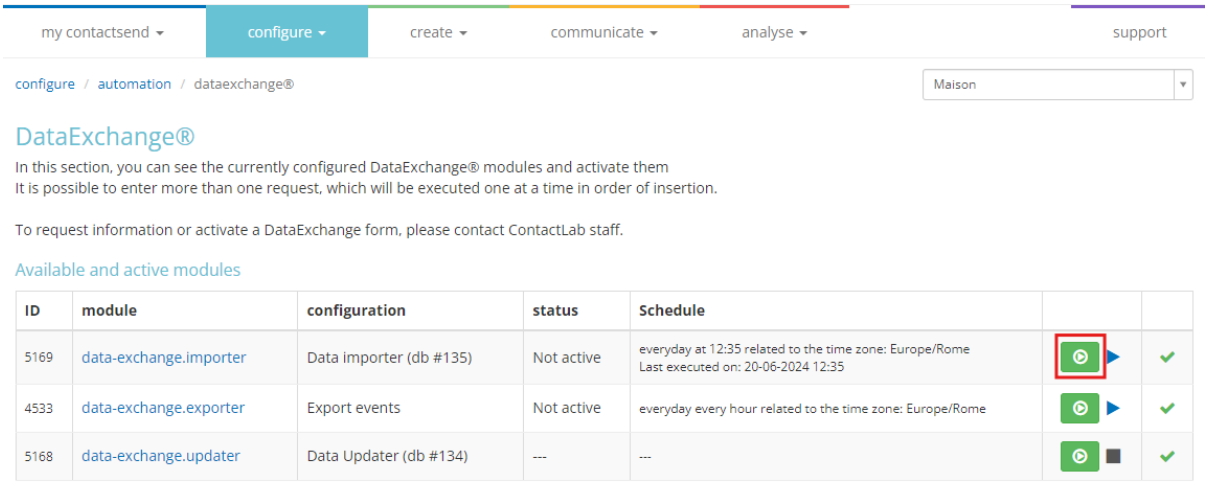
DataExchange modules can be configured and managed using Send web UI. By navigating to “Configure > Automation > dataexchange” a properly authorized user has access to two sections: Configuration (“Available and Active modules”) and log/planning with two tabs (“Available Modules” and “List of items planned or executing”).

The first section, “Available and active modules”:

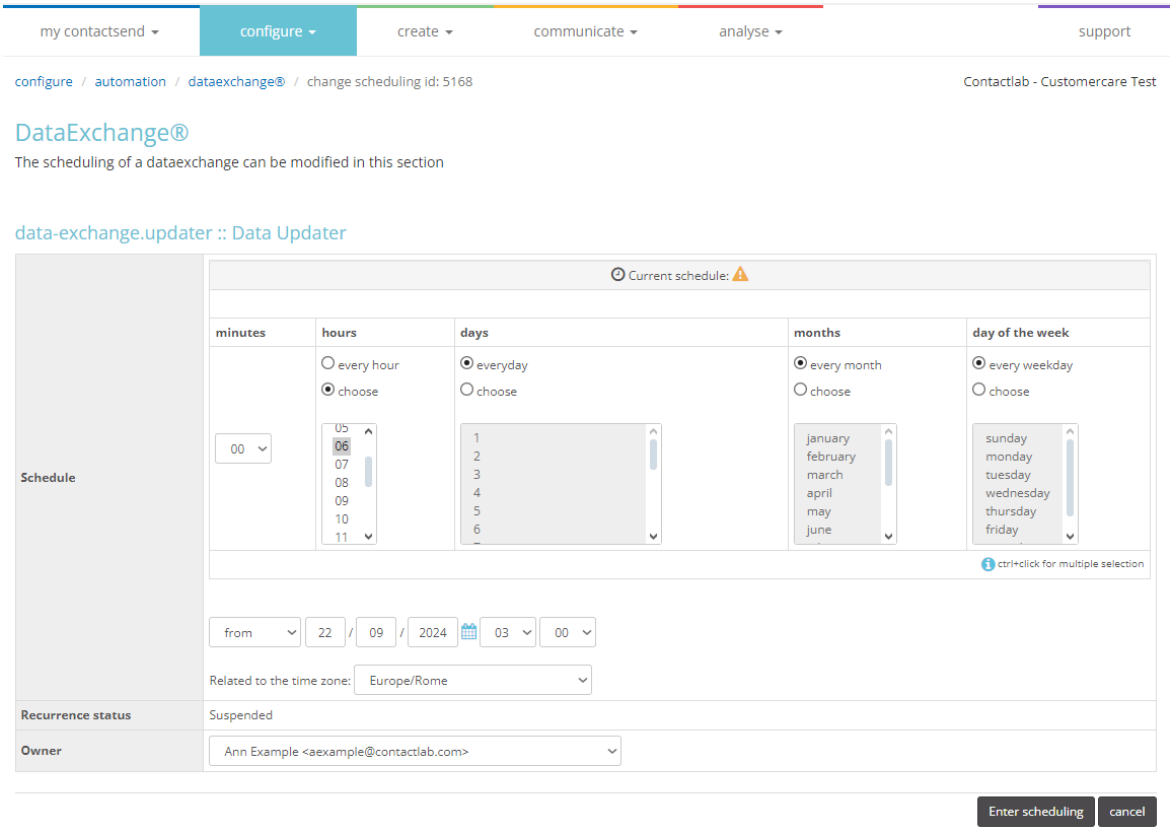
Module identifies the activated module type (DataImporter, DataExporter, DataUpdater…), *Configuration* is a description useful to quickly identify a specific configuration when a large number of modules is active.

In the leftmost field, a *play* icon is shown when the module configuration in the same line will only be executed on demand (i.e. explicitly invoked by user through the web UI). If the icon in the bottom-right corner is a *pause icon* instead, periodic/scheduled execution is configured, and it is possible to suspend it by clicking that icon.

For on-demand modules, execution can be invoked from this same tab by clicking the play icon with a green arrow in its bottom right corner as show in the following picture.



For periodic / scheduled execution configurations (play icon) clicking on the icon will activate the scheduling configuration form:

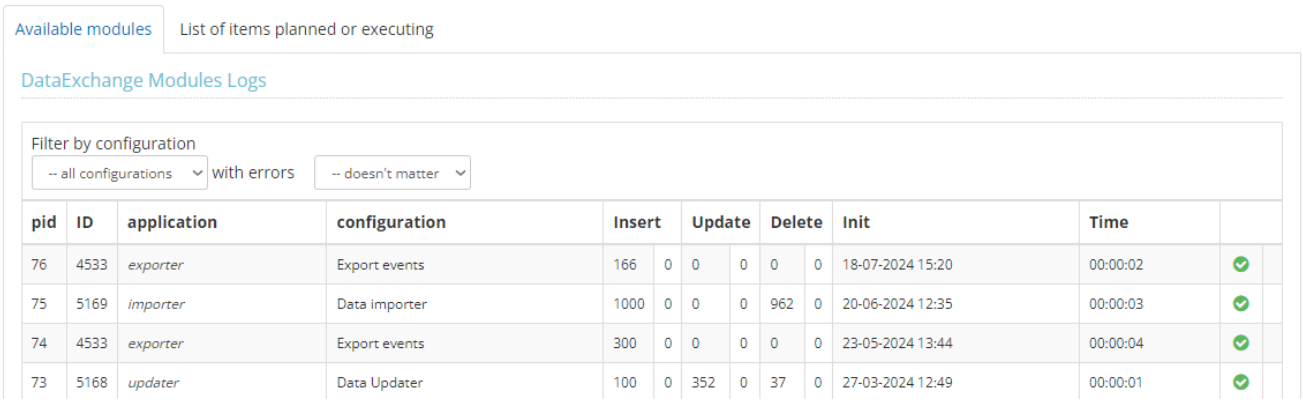


Available options for scheduling are shown: daily, weekly, first or last day of month; or a specific date and time. Some of the options can be combined to ensure maximum flexibility.

Periodic execution can be suspended and reactivated simply by changing the value of the “Scheduling status” listbox.

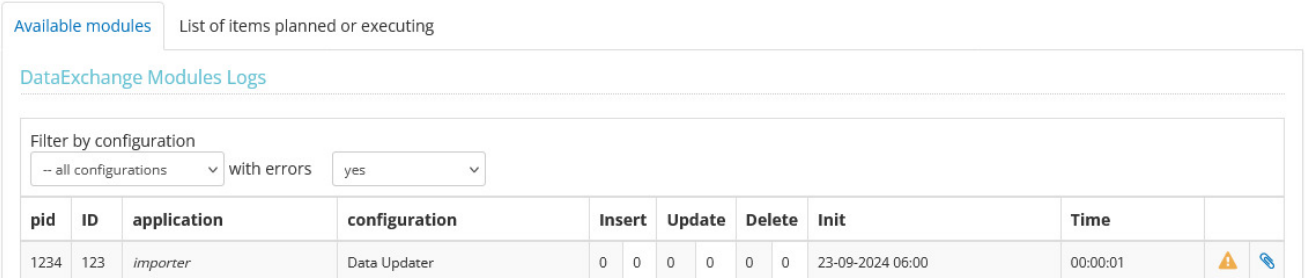
It is also possible to modify the “owner”: the user receiving summary reports for executions of this configuration, and who will be able to monitor execution in real time (with UI notifications).

The second section shows a log of past module executions (“Available modules” tab) or modules scheduled to run in the future or executing at the time this page is invoked (“List of items planned or executing”).



pid	ID	application	configuration	Insert		Update		Delete		Init	Time
76	4533	*exporter*	Export events	166	0	0	0	0	0	18-07-2024 15:20	00:00:02
75	5169	*importer*	Data importer	1000	0	0	0	962	0	20-06-2024 12:35	00:00:03
74	4533	*exporter*	Export events	300	0	0	0	0	0	23-05-2024 13:44	00:00:04
73	5168	*updater*	Data Updater	100	0	352	0	37	0	27-03-2024 12:49	00:00:01

“Filter by configuration” allows showing past requests according to the final result (success or error). When an error or warning occurred, a warning sign icon appears and if the diagnostic log file is available it can be downloaded by clicking on the attachment icon.



pid	ID	application	configuration	Insert		Update		Delete		Init	Time
1234	123	*importer*	Data Updater	0	0	0	0	0	0	23-09-2024 06:00	00:00:01

In the “*List of items planned or executing*” tab, scheduled or currently running configurations are shown.

8. Standard ETL function library

This is a list of ETL functions already available (as an optional configuration) in Send.

Standard ETL functions are applied at the row level i.e. they are configured on specific fields and will execute in the context of a specific row being processed.

Customized ETL functions outside of this list can implement almost arbitrarily complex requirements but will require a specific quotation from your Account/Sales for custom development work.

8.1 Basic::stringDateFormat

Transforms custom date formats to the standard yyyy-mm-dd format.

8.2 Basic::getDomainFromEmailAddress

Extracts the domain part of an email address from a complete email address.

8.3 Basic::getIntFromBoolean

Normalizes true and false values into integers (true: 1, false: 0).

8.4 Basic::trimChars

Returns a copy of the input string, with leading and trailing whitespace removed.

8.5 Basic::force

Unconditionally sets a field/value irrespective of what is specified in the DataExchange input source.

8.6 Basic::integerToBinaryString

Transforms an integer value into a string containing its binary representation (e.g. “6” becomes “110”).

8.7 Basic::semicolonSeparated2bitmask

Creates a bitstring representation out of a sequence of integer values. Requires a constant and configured bistring length.

E.g. suppose we have a 20-positions bitstring representing categories of independent interests for a user profile. The current user profile specifies “1;7;13” as the list of applicable interest categories (1, 7 and 13 are indexes in a fixed list and can mean anything, such as “automotive”, “sports”, and “financial”). This function will generate the string 00000001000001000001 where positions 1, 7 and 13 (right to left) are set to value 1.

8.8 FieldValueMantainer::dontChangeIFFieldValueIS

Enable specifying a “N/A” value in the input source used to mean “do not change value for this field”.

The N/A value is a configured parameter.

E.g. suppose the configured N/A value is “VOID”. If the value “VOID” is found in a column in the input source, the already existing column value will not be touched.

8.9 FieldValueMaintainer::genericTranscode

Transcode specific values in input fields into 1:1 configured values in database.

E.g. for a “gender” column “F” might be transcoded to 0 and “M” to 1.

8.10 LookupTable::checkField

Looks for a value in a lookup table. If found, returns a field in the lookup table for the matching row, or a fixed value. If not found, a fixed value is returned.

8.11 Zone::miscellanea

This is specific to the Italian market. Various “Zone” ETL functions are available to transcode to and from Italian Cities, Provinces and Regions names and ID into each other with geographical bonding rules (e.g. from a City to its containing Region name or ID). IDs are indexes in standard, always available and updated lookup tables in Send; IDs are used to enhance performance of complex planning filters in lieu of descriptive names. Zip codes to Province/Region and Province/Region to Nielsen Area ID transcoding functions are also already available in this subset.

9. DataExchange Post-Actions

A *post-action* is an automatic operation which can be configured to run after a successful DataExchange execution.

The following *post-actions* are available and can be configured on request. Custom configuration requires additional work so please contact your Account or Sales manager in order to receive a quote for the effort.

Additional post-actions can be created on demand with a custom development quotation.

9.1 Chained DataExchange Modules

It is possible to have multiple DataExchange operations configured to start automatically in a chain or sequence. The chain will interrupt at the first “broken link” (a DataExchange module completing unsuccessfully).

9.2 Start a Delivery after a DataExchange Operation

A delivery can automatically be started after a successful DataExchange operation.

9.3 DB Fitness

You can automatically remove recipients that either unsubscribed or were inserted in your blacklist/suppression list.

9.4 Duplicate Remover

After updating your database, you can have this automatic process identify multiple copies of the same email address and remove all but one of them. The one with the highest primary key (usually the most recent) will remain; all other duplicated rows will be removed.

10. Diagnostics messages

Error code	Error message	Possible causes	Possible solutions
DAEX100	Record insertion failed {0}	It usually occurs in the DataImporter and DataUpdater in case of failure of an insertion for sql reasons (value not valid, syntax error, etc.).	Verify if the fields structure of the database in the platform correspond to the one in the file/db source and that the field control (eg. primary key) is correct.
DAEX101	Record updating failed {0}	It usually occurs in the DataUpdater in case of failure of an updating for sql reason (value not valid, syntax error..) or because the updating request didn't find the requested record.	Verify that the fields structure of the database in the platform correspond to the one in the file/db source and that the field control (eg. primary key) is correct. Verify that the user is present in the db of the platform: if not contact our Customer Care.
DAEX102	Record elimination failed {0}	It usually occurs in the DataUpdater in case of a cancellation failure for sql reasons (value not valid, syntax error..) or because the updating request didn't find the requested record.	Verify that the fields structure of the database in the platform correspond to the one in the file/db source and that the field control (eg. primary key) is correct. Verify that the user is present in the db of the platform: if not contact our Customer Care.
DAEX104	Control code not found using the index {0}	It usually occurs in the DataUpdater when the code that decides if carry out with the insert, update or delete is not found in the dedicated column that should host this specific value (operation field).	Verify that the operation field is present and that it contains the correct values. The values have been defined in the activation form that is filled in by the customer during the setup stage and that is been given back to his/her Account.
DAEX105	Number of indexes allowed into the line out of range: {0}	It usually occurs in the DataUpdater or DataImporter with ETL active system when the number of columns that are in the file are different from the ones told in the configuration.	Contact Customer Care.
DAEX150	SQL error, message received: {0}	It usually occurs in all DataExchange when SQL exceptions are provoked and that are not referable to a problem.	Contact Customer Care.
DAEX200	User db n. {0} seems to be busy by campaign {1} in status {2}	It usually occurs in all DataExchange when their execution is done on their user db that is currently in use in that specific moment (while a campaign is running out). The message also shows the Id of the campaign that keeps the user db busy and the status of the sending.	If there is a campaign outgoing, wait until the end. If the deployment is blocked in the "Communicate" area or if there is no campaign outgoing, contact Customer Care.
DAEX201	The Contactlab user db is not available at the moment, message received {0}	It usually occurs for all components of the DataExchange when de user db in the platform is not available (for example for temporary connection problems).	Contact Customer Care.

DAEX301	Source not available, message received: {0}	It usually occurs when the DataImporter or the DataUpdater doesn't find the source file (csv/xml) or it is not able to read it for authorization reasons.	Verify that the file is in the path and that it has the name that has been specified in the activation form filled in by the customer during the set-up stage and that has been given back to his own Account.
DAEX304	The remote authentication failed, message received: {0}	This message occurs when a DataExchange tries to enter into an external source, but the login details (username and password) are incorrect.	Verify that the file is in the path and that it has the name that has been specified in the activation form filled in by the customer during the set up stage and that has been given back to his own Accountant and that the login details to the sftp space (if managed by the customer) are correct.
DAEX305	Connection attempt failed, message received: {0}	It occurs when a DataExchange tries to enter into an external source, but the communication with the source server expires. It can depend on the missing authorization via firewall from the customer side to the IP from which Send is connected to or because the source service is not active.	If the ftp space is managed by the customer, verify with your own technical department that everything is working properly. If everything is correct but the problem is still there, contact our Customer Care.
DAEX306	There was a problem when attempting to delete/remove source files, error message: {0}	An error occurred when DataCollector tried to delete or rename the external file source.	Typically caused by permission problems on the folders where affected files are located.
DAEX307	There was a problem when attempting to delete/remove the semaphore file, error message:{0}	An error occurred when Poller tried to delete or rename the semaphore files.	Typically caused by permission problems on the folders where affected files are located.
DAEX900	Error not classified {0}	It usually occurs in case of not classified errors that are not present in the ones listed.	Contact Customer Care.
DAEX901	Problem during the sending of the report, message received: {0}	This message occurs in case of problems in communicating the report via email. The addresses shown as addresses of the report could not be spelled correctly.	Contact Customer Care.
DAEX999	Process interrupted {0}	It can depend on the excessive last of the DataExchange process due to unknown reasons so it remains "dangling".	Contact Customer Care.

11. Frequently Asked Questions & Answers

You have too many options available. What is your suggested configuration?

Our suggested configuration for any DataExchange is:

- Primary key (USER_PK) managed by you

This configuration, although a little more difficult to set up, will guarantee the most flexibility and will grow with your databases. Millions of updates a day are not an issue if performed as updates with a DataUpdater; while refreshing a database of millions of recipients (such as with a DataImporter) will sooner or later become too slow.

By keeping control of your primary key you can track events for your recipients across the history of their relationship with you. Email addresses change – that's a fact of life. If you do not recognize this, you will sooner or later run into problems.

Can I use encryption to protect data transferred to Send?

Send natively supports public/private key encryption with PGP o GPG.

How can a primary key be managed in Send?

Send natively supports an external (customer-managed) primary key in any user database, provided it is a positive integer. We only support positive integer for performance reasons; it is not possible to use a differently typed key. And we do not support email addresses as primary keys for a number of reasons (the most important one being: people change email addresses more frequently than one could think – and we're quite in a good position to know this for sure).

If you have a positive integer primary key or if you can build a synthetic key from your own, by systematically including it in DataExchange datasource, you will ensure Send view of your recipients will be consistently aligned with your own. For instance, events for a given recipient in multiple deliveries will be currently tracked to the same recipient. Primary key consistency, moreover, enables a quite efficient use for DataUpdater allowing small and frequent synchronizations even with huge recipient databases.

A primary key must by definition be unique in a given user database: it is not necessarily unique across all user databases.

If you do not have a suitable primary key in your origin datasource, you can have Send managed it automatically; Send will create a new primary key for every new record you add to your user database (once again, primary key is unique in a given user database). However, since Send has no knowledge of what your primary key is, if you add multiple copies of the same recipient, they will be identified by different primary keys. Send will be unable to help you track events across deliveries (across updates to the database).

Customizations of the DataExchange modules can be implemented to delegate to Send maintenance of the relationship between your primary key (when it does not fit an integer, possibly even when composed by more than one field). If you are interested in an evaluation of this approach contact your Account.

Why is my User Database locked after a failed DataExchange operation?

In Send we follow a simple principle: contrary to web pages, once an email has been sent, it cannot be corrected nor recalled. We believe sending a wrong email or sending an email to a recipient who doesn't want it is a much more serious incident than failing to send an email. So when a DataExchange operation fails, our assumption is your User Database failed some important updates and we leave it "locked" – no delivery can be sent when the database is locked. A delivery will retry itself for some time, so if you repeat the DataExchange operation and this time it completes successfully, it will proceed. After some time it will enter a failed state.

You can ask your DataExchange configuration to ignore some error. This requires manual configuration only because we want to discuss with you to make sure you are well aware of the dangers.

Can I have more than one DataExchange configuration active on the same User Database?

Yes, you can. Schedule them carefully to avoid overlaps and remember the User Database remains locked during each execution. Execution will be serialized, in order to preserve consistency of data. You can also "chain" DataExchange configurations so that they executed in a given sequence without risking interlocking issues.

Can I automatically add a new row when a DataUpdater 'U' operations fails because the row is not there?

Yes, you can have your DataUpdater configured so it will automatically transform an U (update) instruction to I (insert) if the master condition for that row is not matched.

How long are detailed data on events available?

Detailed data (including personal information such as email address of the recipient and all events) are available for 12 months. After this period, they are aggregated so that summary info and indicators on campaigns are available until customer can access Send, but details are deleted permanently. This ensures compliance with the current regulations and best practices on personal data protection for marketing use. Using a DataExporter feed, you can receive all event details and archive them with a different retention policy, if you have the need. It is possible to ask for a shorter retention period, however we will not accept a retention period shorter than 90 days to ensure some of the mandatory unsubscribe mechanisms work for at least this length of time.

12. Contacts

The Contactlab Customer Care is always ready to examine your questions. Just open a new ticket from our web UI (Support section), on our support website https://support.contactlab.com, or send an email to helpdesk@contactlab.com. Customer with Premium support levels can also call the special Premium Support phone number.