

DBPLUS

# DBPLUS Data Replicator User's Guide

Last modified: 2019-11-25  
Product version: 2019.4.2

## Table of contents

1.	Introduction to DBPLUS Data Replicator .....	4
1.1	What is DBPLUS Data Replicator? .....	4
1.2	Product architecture and important facts.....	4
	Oracle as source database.....	4
	SQL Server as source database.....	5
	All configurations.....	6
1.3	Definitions .....	6
2	Installation and configuration .....	7
2.1	Installation.....	7
2.2	Windows services .....	7
2.3	Replication manager configuration .....	7
	GUI Client Service .....	8
	REST Service.....	8
3	Starting the program .....	10
4	Creating a new replication .....	11
4.1	Connection setup .....	12
	Oracle .....	13
	SQL Sever.....	14
	Additional configuration for SQL Server.....	15
	PostgreSQL .....	17
4.2	Replication rules setup.....	17
	EXACT REPLICATION .....	18
	CUSTOM REPLICATION .....	19
	Database level .....	22
	Schema level.....	22
	Table level.....	24
	Tablespace mapping.....	27
	Wildcards.....	27
	Finalizing configuration of replication rules .....	28
	Replicated DDL commands depending on source and target databases.....	28
4.3	Generate and view report .....	28
4.4	Start options .....	29
4.4.1	Manual mode .....	29
4.4.2	Automatic mode.....	31
4.5	Name and create the replication.....	31

5	Starting replication .....	33
5.1	Prerequisites.....	33
	List of prerequisites for Oracle databases.....	33
	List of prerequisites for SQL Server databases.....	33
	List of prerequisites for PostgreSQL databases.....	33
5.2	Artefacts .....	34
6	Replication monitoring and management .....	35
6.1	Replication states .....	35
6.2	Statistics.....	36
6.3	Table level monitoring and management .....	37
6.4	Alerts .....	40
	Missing prerequisites .....	41
	Errors encountered when applying changes in the target database (applier errors).....	41
6.5	Automatic email notifications concerning replication problems.....	42
6.6	Configuration edit.....	44
	Start point and end point settings.....	45
	Basic configuration .....	45
	Advanced configuration .....	49
	Data conversion and data type mapping rules.....	54
	Configuration of optional alerts .....	57
	Planned replicator breaks schedule .....	57
6.7	Maintenance tools.....	61
	Tool for generating scripts to transfer data using data pump .....	62
	Tool for viewing the replication preparation report .....	65
	Change Capture Component maintenance tool.....	65
	Monitoring change capture job status .....	67
6.8	Restarting replication from a new start point.....	68
6.9	Viewing the logs .....	68
6.10	Deleting replication .....	69
7	Additional information .....	71
7.1	Information about replication status accessible in the source database .....	71
7.2	Database level API for managing planned breaks.....	72
7.3	Manually overriding server addresses of Always On Availability Group nodes.....	73
8	Restrictions and limitations.....	74
8.1	Products.....	74
8.2	Oracle as source database.....	74

8.2.1	Supported data types (source database) .....	74
8.2.2	Restrictions for CUSTOM mode.....	75
8.2.3	Restrictions for EXACT mode.....	75
8.2.4	Ignored schemas.....	75
8.1	SQL Server as source database.....	76

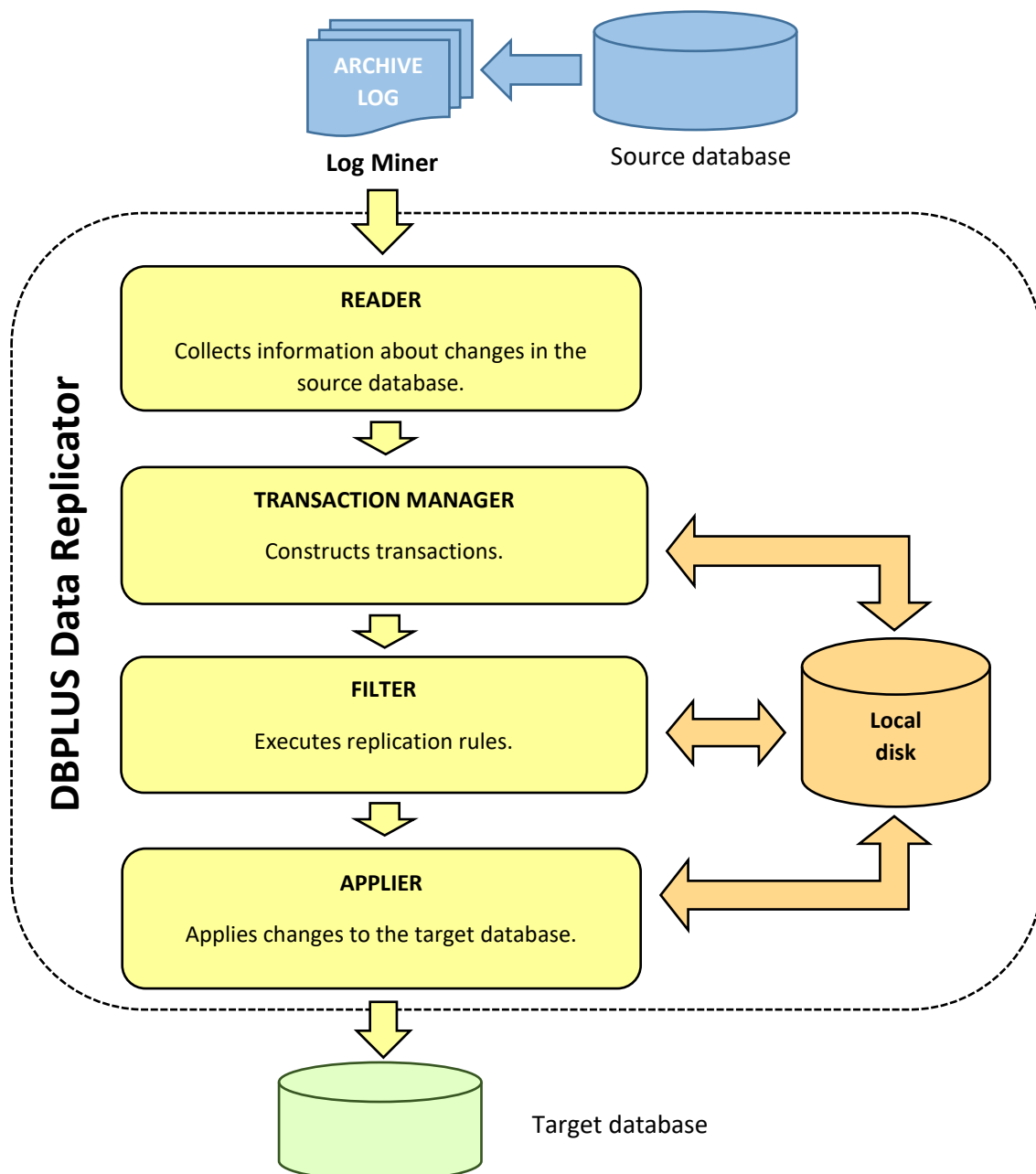
## 1. Introduction to DBPLUS Data Replicator

### 1.1 What is DBPLUS Data Replicator?

Data replicator continuously maintains data in one database (called target database) as a copy of data in another database (called source database). Data in the target database is updated online (with a small delay) using information about changes performed in the source database. User can select objects (schemas, tables) which are subject to replication and determine which types of changes are replicated.

### 1.2 Product architecture and important facts

Oracle as source database



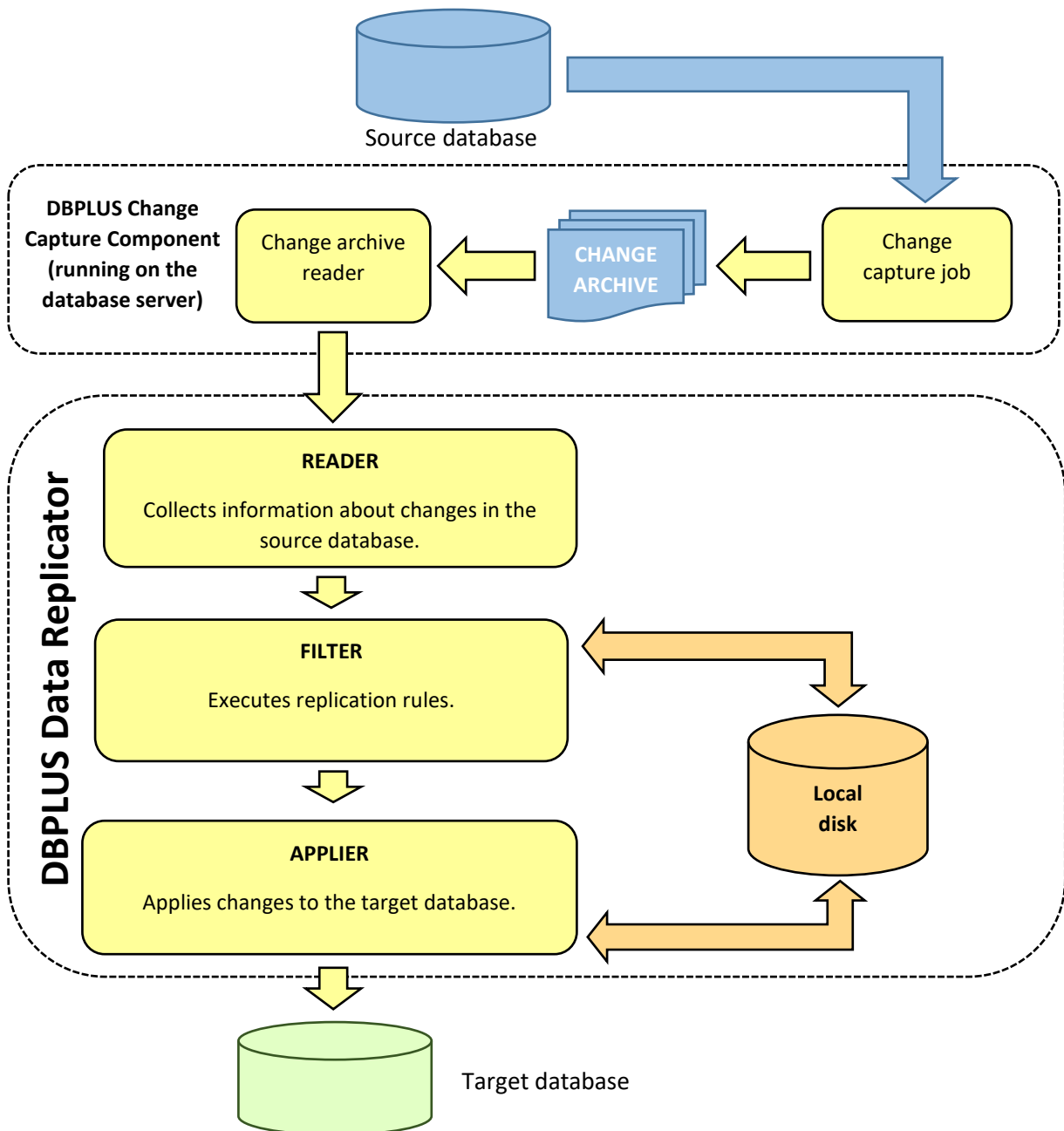
*DBPLUS Data Replicator does not interfere with the working of the source database*

Information about changes the source database undergoes is collected by reading database transaction logs with the help of LogMiner tool, which is available on Oracle database systems. The only impact replicator has on the source database is a slight load which is generated by LogMiner activity.

*Replicator is prepared to handle transactions of arbitrary size, the only condition being adequate space is available on the hard disk.*

Data is kept in temporary files on disk when needed. Because these files can take up a lot of space user has an option to [set the location of replicator's temporary files folder](#).

SQL Server as source database



Replications with SQL Server as the source database make use of CDC (Change Data Capture), which is part of the standard functionality included in the SQL Server product line. Component which supports the change capture process is installed on the server instance where the source database resides, the component is a database containing a number of ddl libraries along with procedures and functions. Change capture process acquires information about changes occurring in databases of interest (using CDC) and writes them to files within the change archive. These information is then read by the replicator.

All configurations

*Replicator is equipped with mechanisms which prevent executing the same operation twice as well as "losing" operations (never executing them). These mechanism work both in the case of orderly replication shutdown (stopping the service) and an abrupt stop (e.g. system crash).*

*After replication is orderly shutdown or abruptly stopped (e.g. system crash) processing resumes in the same place where it has stopped or with some small (a few minutes' worth) overhead.*

Replication state snapshots performed every few minutes together with the temporary data stored on the disk allow replicator to resume its work without a need for processing the same changes from the source database for the second time.

If temporary data or replicator state snapshot gets corrupted or lost, then information about changes in the source database must be processed anew in order to construct complete image of all transactions which have not yet been applied on the target database. In such a situation the best course of action is to determine the first SCN number which has to be considered and set it in [replication options](#) as the new replication start point.

*Current replicator version can use only Oracle database as a source database. The following database systems are supported as a target database: Oracle, SQL Server and PostgreSQL.*

## 1.3 Definitions

<b>Source database</b>	A database from which data (and data structure) is copied by the replication process.
<b>Target database</b>	A database in which data (and data structure) is updated by the replication process.
<b>Replication manager</b>	Service which provides ability to create, configure and delete replications and to monitor and manage replication processes on the machine on which it's installed.
<b>Replication</b>	Description of how data in target database is to be updated with changes from source database along with the current state of the process.
<b>Replication process</b>	Process of updating target database with changes from source database. When talking about starting (or stopping etc.) replication process we will also use a shorter form "starting replication".

## 2 Installation and configuration

### 2.1 Installation

The installation program allows to install one or more of the following components:

- **DBPLUS Data Replicator**  
The core functionality of data replicator.
  - **Server**  
Windows service which allows to create and manage replication processes. We will call this service „replication manager”.
  - **Client**  
Provides rich client GUI interface for managing replication locally (on the machine it has been installed) or remotely.
- **DBPLUS Data Transfer**  
Tool for transferring data between tables in relational databases.
  - **Manager**  
Part of the tool which handles reading from source database and management of data transfer task.
  - **Writer**  
Part of the tool which handles writing data into the target database.

As of this document we will only cover DBPLUS Data Replicator related topics, the DBPLUS Data Transfer tool will only be mentioned in context of its use by the replicator.

### 2.2 Windows services

Replication manager is visible on the Windows service list as „DBPLUS data replication manager”. Each replication process is run as a separate Windows service visible as „DBPLUS data replication (replication name)”. Replication manager and services executing replication processes can be started and stopped using standard service management tools available on the Windows operating system.

Replication manager service is run as *local system*.

Services executing replication processes are run as *local service*.

### 2.3 Replication manager configuration

To configure replication manager run the configuration program available in the Windows menu under DBPLUS → DBPLUS Replication Manager Configuration on the machine where the service is installed.



## GUI Client Service

This section allows you to configure the service used by the GUI client described in this document to access functions .

### Authentication

GUI Client Service authentication uses authentication objects (users, passwords, groups) of the Windows system. You can choose between authentication on the local machine and domain authentication. When using domain authentication you can specify domain name. By default the domain on which replication manager service is run will be used.

### Access

By default every user who can log in to given machine/domain can access the replication manager. In order to restrict access select option **Restrict access to a specific group** and enter name of Windows user group into **Group** field.

### Network setup

By default service used by GUI client is accessible as a network service on port 5220. If port 5220 is taken please enter the number of an available TCP/IP port.

## REST Service

This section allows you to configure service which provides a number of replication manager's function by means of REST technology. The REST service provides basic functionality concerning replication management and is designed for use in scripts. By default the REST service is *disabled*.

Details concerning REST API are discussed in [a separate document](#).

### Authentication

REST Service functions can be made accessible without restrictions or protected by password, in the latter case the username and password must be supplied.

### *Network setup*

By default REST Service is accessible as a network service on port 5220. If port 5220 is taken please enter the number of an available TCP/IP port.

### 3 Starting the program

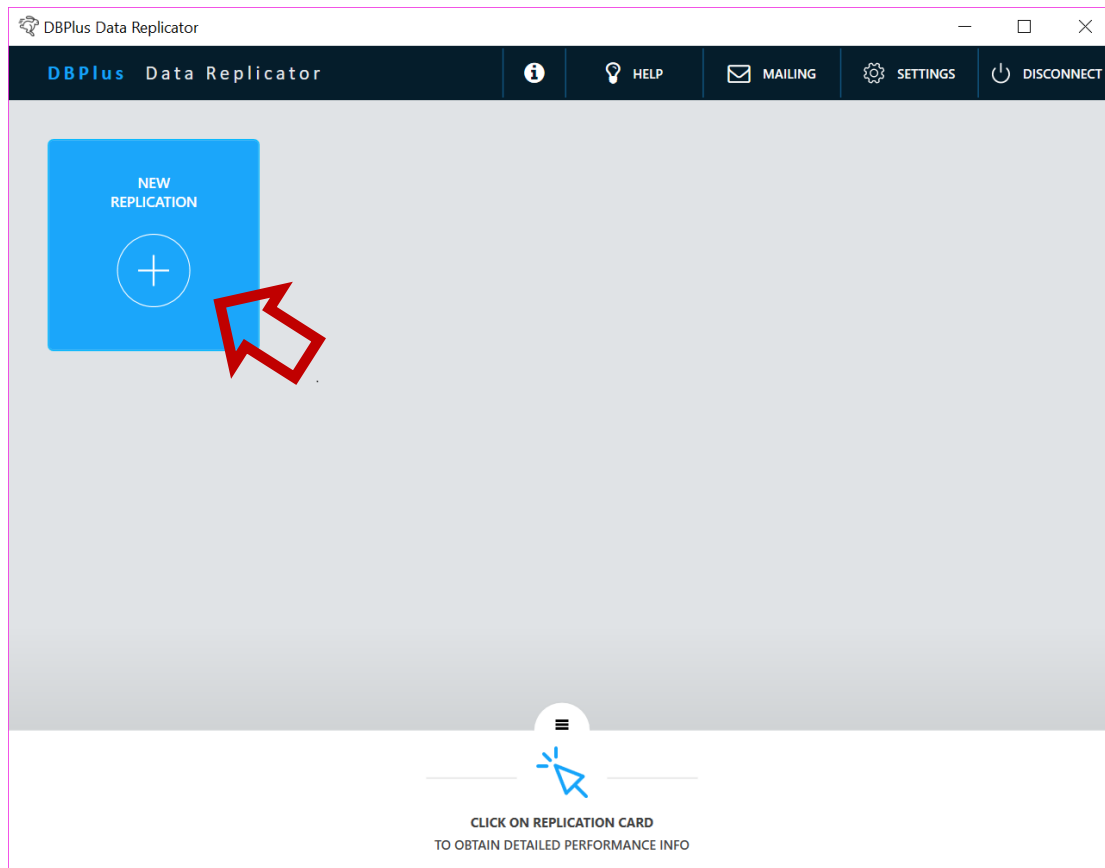
Replication management GUI can be opened from Windows menu under DBPLUS → DBPLUS Data Replicator or using desktop shortcut.

The **port** field is to be left empty unless [network options have been changed](#).

Windows authentication can be used only if the machine you are using is logged in to the Windows domain.

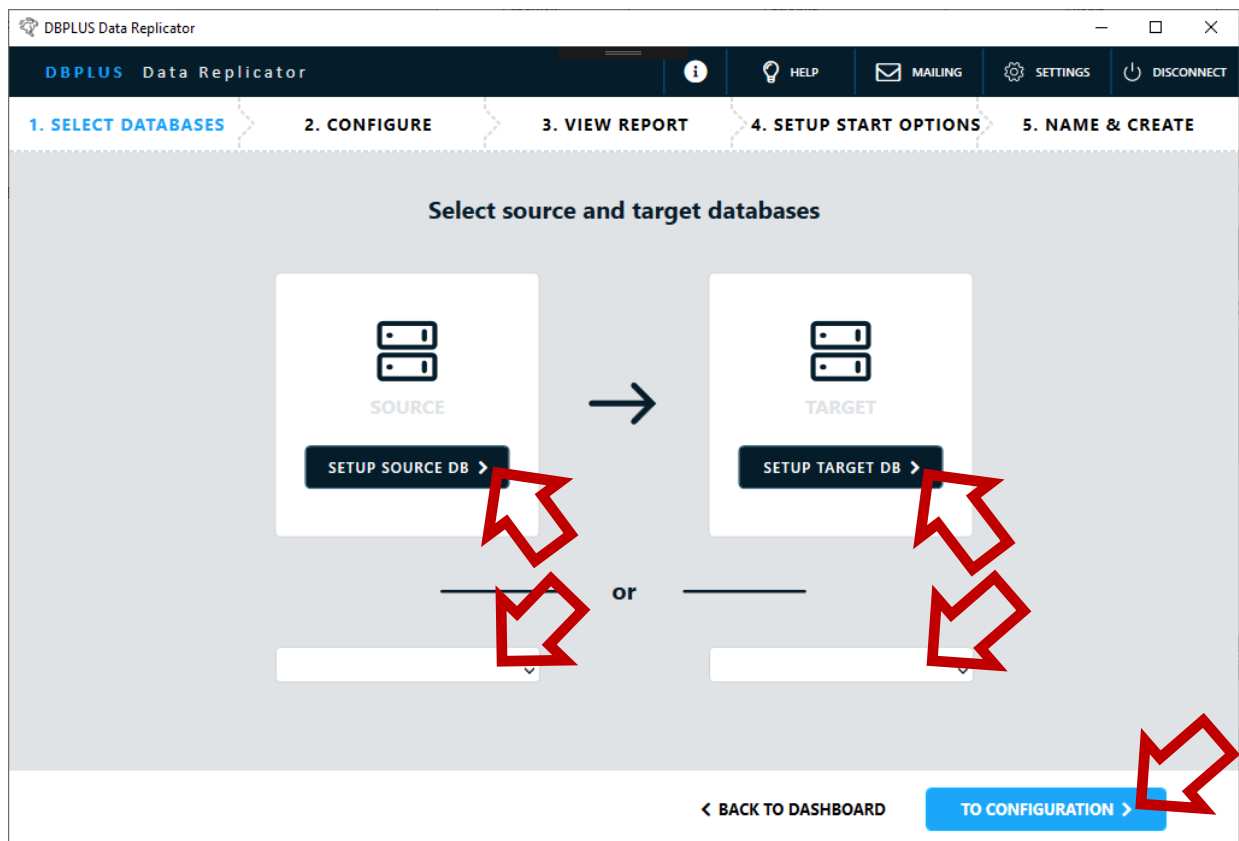
## 4 Creating a new replication

After we have connected for the first time only **NEW REPLICATION** button will be visible on the main screen.



Click on it to move on to entering configuration for the new replication.

## 4.1 Connection setup



At this step connection to the source and target databases is configured. Click **SETUP SOURCE DB** or **SETUP TARGET DB** to enter settings from scratch or reuse settings that have been entered earlier by selecting from the drop down lists below. After selecting settings from a drop down list you can use **SETUP SOURCE DB** or **SETUP TARGET DB** to verify or edit them.

After configuring source and target connections is completed, move to the next stage by clicking on **TO CONFIGURATION** button.

DBPLUS Data Replicator

DBPLUS Data Replicator

1. SELECT DATABASES 2. CONFIGURE 3. VIEW REPORT 4. SETUP START OPTIONS 5. NAME & CREATE

**Setup source database connection** ☒ save for later use

☒ ORACLE ☐ MS SQL SERVER

connection name: SOURCE

setup mode: Basic

server: myserver

port: 1522

service identifier type: Service name

service name: SOURCE

container name:

☐ Setup replication user manually  
☒ Let program create and configure the user for you

sys admin credentials (for setup purposes only)

user name: SYS

password: ...

replication user credentials ☒ create if not exists

user name: MYUSER

password: ...

SETUP REPLICATION USER >

< CANCEL SAVE >

In the **connection name** field enter name for your connection. The name you enter here will be later used by the program to describe the connection. If **save for later use** is checked the settings you have entered will be available for reuse in a drop down list.

There are two ways to setup user for the replication process:

### Setup replication user manually

Enter name and password of a user already created and configured.

If this option has been selected, **TEST CONNECTION** button will be available. Use this button to check if connection works. Remember that connection is verified *on the machine where the service is run*, not on the machine on which you are currently working.

### Let program create and configure the user for you

User is automatically created (provided that **create if not exists** option is checked) and configured. For this purpose system administrator credentials must be provided. System administrator credentials are used for the duration of configuration process only and are not stored.

If this option has been selected, **SETUP REPLICATION USER** button will be available. Clicking on it will start a procedure which will create (if needed), and grant necessary privileges to, the replication user. Successful execution of this function also verifies that connection works.

When finished, accept settings by clicking on **SAVE** button.

## SQL Sever

DBPLUS Data Replicator

1. SELECT DATABASES 2. CONFIGURE 3. VIEW REPORT 4. SETUP START OPTIONS 5. NAME & CREATE

**Setup source database connection** ☒ save for later use

☐ ORACLE ☒ MS SQL SERVER

connection name: SOURCE

server: myserver

port: 1433

service name:

database name: MyDB

☐ require secure connection (SSL)  
☐ trust server certificate

☐ Setup replication user manually  
☒ Let program create and configure the user for you

sys admin credentials (for setup purposes only)

authentication: Database

user name: sa

password: ...

replication user credentials ☒ create if not exists

authentication: Database

user name: MYUSER

password: ...

SETUP REPLICATION USER >

< CANCEL SAVE >

In **connection name** field enter name for your connection. The name you enter here will be later used by the program to describe the connection. If **save for later use** is checked the settings you have entered here will be available for reuse in a drop down list.

There are two ways to setup user for replication process:

### Setup replication user manually

Enter name and password of a user already created and configured.

If this option has been selected, **TEST CONNECTION** button will be available. Use this button to check if the connection works. Remember that connection is verified *on the machine where the service is run*, not on the machine on which you currently work.

### Let program create and configure the user for you



User is automatically created (provided that **create if not exists** option is checked) and configured. For this purpose system administrator credentials must be provided. System administrator credentials are used for the duration of configuration process only and are not stored.

If this option has been selected, **SETUP REPLICATION USER** button will be available. Clicking on it will start function which will create (if needed), and grant necessary privileges to, replication user. Successful execution of this function verifies that the connection works.

By default only packets associated with logging on are encrypted. Use **require secure connection** option to enforce encryption of data. If **trust server certificate** option is enabled server certificate will not be verified, which means you may use self-signed certificate.

If the source database belongs to an Always On Availability Group user should be configured on all servers belonging to the group, otherwise replication will stop after failover switch. If automatic user creation and configuration option has been selected program will suggest executing this operation on all servers belonging to Always On group. If necessary user configuration on specific servers may be turned off using the **include** check box.

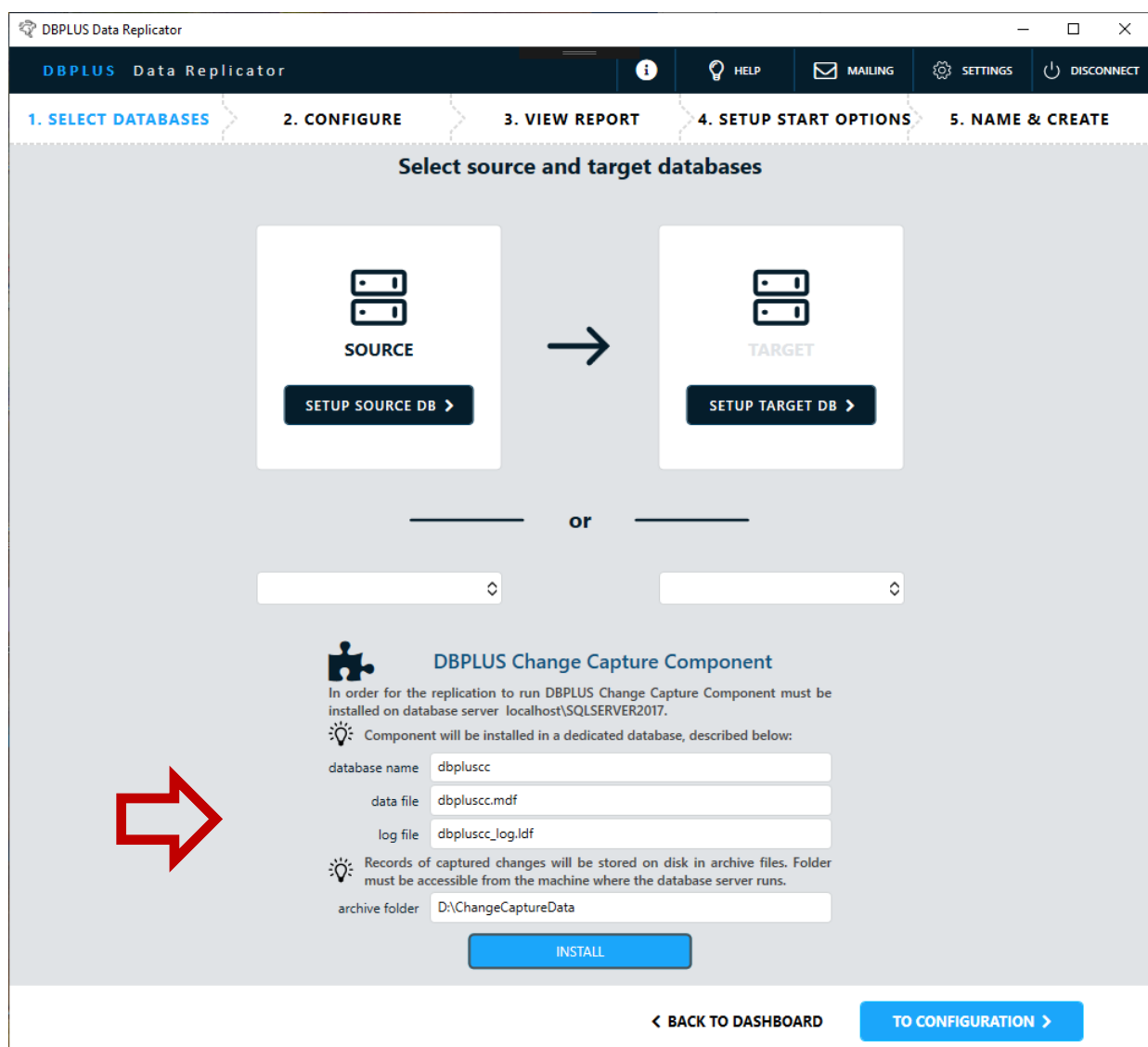
The selected database belongs to **Always On Availability Group** AlwaysOnCluster. Replication user should be set up on all nodes of the group to ensure correct functioning of replication process after switch occurs.

 DBPLUSSQLSRV1\SQLNODE1	<input checked="" type="checkbox"/> include
 DBPLUSSQLSRV2\SQLNODE2	<input checked="" type="checkbox"/> include

If replication manager is unable to connect to database servers using names displayed here (DBPLUSSQLSRV1\SQLNODE1 or DBPLUSSQLSRV2\SQLNODE2 in the above example) please apply solution described [here](#).

When finished, accept settings by clicking on **SAVE** button.

### Additional configuration for SQL Server



DBPLUS Data Replicator

DBPLUS Data Replicator

1. SELECT DATABASES 2. CONFIGURE 3. VIEW REPORT 4. SETUP START OPTIONS 5. NAME & CREATE

Select source and target databases

SOURCE TARGET

SETUP SOURCE DB > SETUP TARGET DB >

or

database name dbpluscc

data file dbpluscc.mdf

log file dbpluscc\_log.ldf

archive folder D:\ChangeCaptureData

INSTALL

< BACK TO DASHBOARD TO CONFIGURATION >



Upon returning to the previous screen you will notice that a DBPLUS Change Capture Component installation panel now appears at the bottom part. You can learn more about the component [here](#).

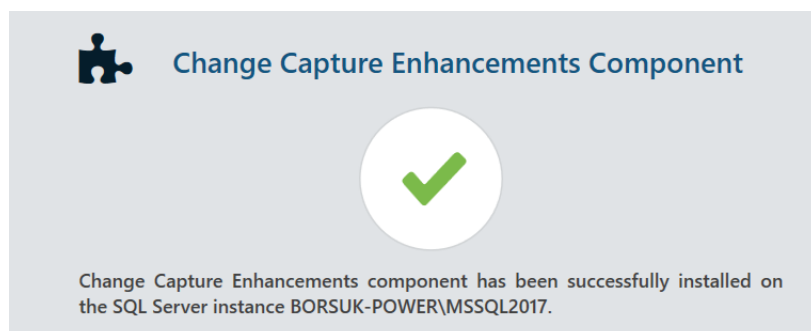
The following parameters can be set:

database name	Name of the database where the component will be installed.
data file	Name of data file for the database where the component will be installed.
log file	Name of transaction log file for the database where the component will be installed.
archive folder	Change archive folder location. In the process of capturing changes using the component change information is written to disk. Repository of change information stored in files is called „change archive“. The parameter specifies the archive's location. Remember this location <i>must be accessible for the SQL Server instance</i> .

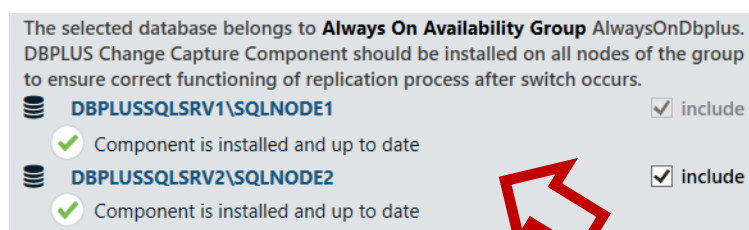
Only basic parameters (necessary to install the component) can be set here. Advanced options of DBPLUS Change Capture Component can be edited after replication has been created using [Change Capture component management tool](#).

After parameters have been set, execute installation by clicking the **INSTALL** button.

After successful installation the following information will be displayed:



If the source database belongs to an Always On Availability Group component should be installed on all servers belonging to the group, otherwise replication will stop after failover switch. By default installation and upgrade is executed on all servers in the Always On group. If necessary installation on specific servers may be turned off using the **include** check box. Installing the component on the primary replica is required for the replication configuration process to continue.



If replication manager is unable to connect to database servers using names displayed here (DBLPUSSQLSRV1\SQLNODE1 or DBLPUSSQLSRV2\SQLNODE2 in the above example) please apply solution described [here](#).

## PostgreSQL

DBPLUS Data Replicator

DBPLUS Data Replicator

1. SELECT DATABASES 2. CONFIGURE 3. VIEW REPORT 4. SETUP START OPTIONS 5. NAME & CREATE

Setup target database connection ☒ save for later use

☐ ORACLE ☐ MS SQL SERVER ☒ POSTGRES

connection name POSTGRES\_1

server myserver

port 5432

service name

database name MYDB

replication user credentials

user name MYUSER

password

☒ require secure connection (SSL)  
☐ trust server certificate

TEST CONNECTION >

< CANCEL SAVE >

In **connection name** field enter name for your connection. The name you enter here will be later used by the program to describe the connection. If **save for later use** is checked the settings you have entered here will be available for reuse in a drop down list.

Secure connection is always used if the server supports it, use **require secure connection** option to force using secure connection. If **trust server certificate** option is enabled server certificate will not be verified, which means you may use self-signed certificate.

When configuration is completed, you can click on **TEST CONNECTION** button to check if the connection works. Remember that connection is verified *on the machine where the service is run*, not on the machine on which you currently work.

When finished, accept settings by clicking on **SAVE** button.

### 4.2 Replication rules setup

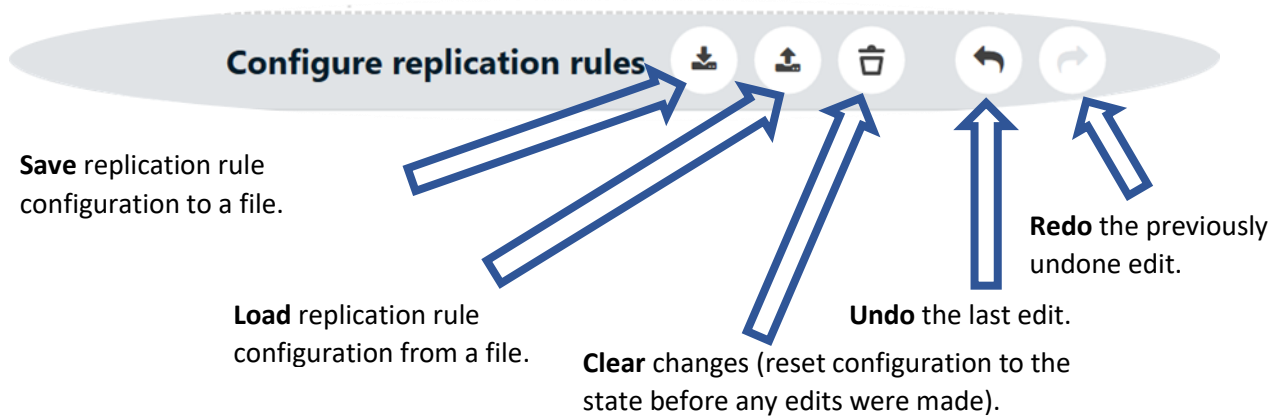
At this stage we specify **what** and **how** to replicate, that is

- what schemas and tables,
- what operations

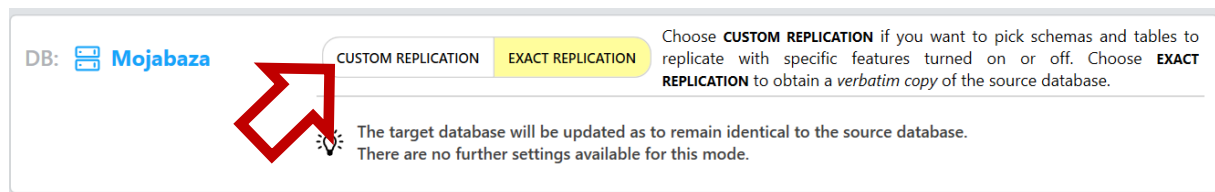
are to be replicated, and what **transformations** are to be applied:

- changing of schema names,
- changing of table names.

Replication rules setup provides the following toolbar:



The first decision you have to make is the choice between **CUSTOM REPLICATION** mode and **EXACT REPLICATION** mode.



## EXACT REPLICATION

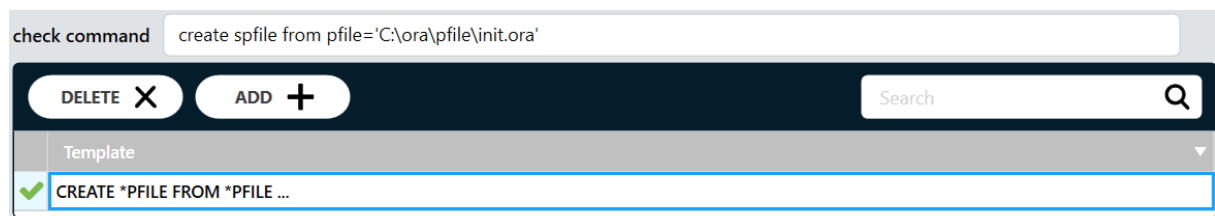
Replication of the whole database – maintaining the target database as a verbatim copy of the source database. If this mode is selected no further setup is required.



The *exact* mode is only accessible when replicating from one Oracle database to another Oracle database.

Changes concerning some predefined system schemas will not be replicated. There are some [restrictions and limitations](#), in particular some data types are not supported.

In the **Exclude specific DDL commands** section you can specify DDL commands, which will not be replicated. This is useful to exclude administrative commands which are dependent on system environment.



The following rules are in effect when defining DDL command template:

- Tokens (ie. keywords, operators, literals) of a template are compared with tokens of a DDL command, whitespace separate tokens, number and kind of whitespace characters does not matter
- Upper/lower case does not matter
- Comments in the analyzed DDL command are ignored

- Within a token you can use wildcards \* and ? (arbitrary string of characters and a single arbitrary character)
- Special token ... (three dots or ellipsis) matches arbitrary sequence of tokens (possibly empty) in the analyzed DDL command

In order to verify whether templates work as intended enter a DDL command into the **check command** field. A green sign will be displayed next to templates which match the command.

## CUSTOM REPLICATION

Replication of selected operations for selected schemas and tables. In this mode the following entities can be replicated:

- data
- data structures: tables (with indexes) and views
- code: packages, procedures and functions



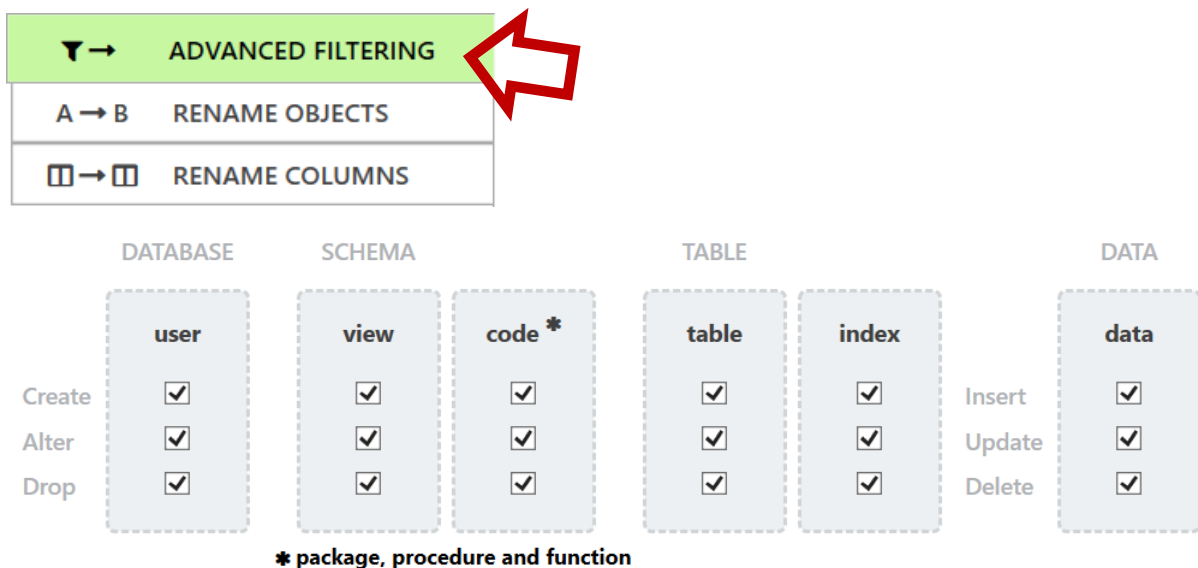
There are some [restrictions and limitations](#), in particular some data types are not supported.

Configuration of the *custom* mode consists of three levels:

1. Database (global) level
2. Schema level (for specific schemas)
3. Table levels (for specific tables)

### Advanced filtering options

On each of these levels you can set **advanced filtering** options which control replication of specific **operations** performed on specific database **objects**.



Operations concerning whole database:

<b>user</b>	<b>Database users.</b>
-------------	------------------------

Operations concerning specific schemas:

<b>view</b>	<b>View definitions.</b>
-------------	--------------------------

<b>code</b>	Packages, procedures and functions.
-------------	-------------------------------------

Operations concerning specific tables:

<b>table</b>	<b>Table definitions.</b>
<b>index</b>	Index definitions.
<b>data</b>	Data (records).

### Object rename options

On the schema and table levels you can set object **rename** settings. Object rename settings allow to replicate data between schemas and tables when their names differ between the source and target databases.

ADVANCED FILTERING

RENAME OBJECTS

RENAME COLUMNS

CHANGE ☒

TARGET SCHEMA NAME

PRODUCTS

CHANGE ☒

TARGET TABLE NAME

P\_<NAME>

Use <NAME> as a placeholder for the original schema or table name. E. g. to insert prefix ALT\_ before the names of all tables in a schema specify ALT\_<NAME> as table name replacement on the schema level.

When specifying name of the target object you can use special character sequence <NAME>. This sequence will be replaced with the name of the source object. This makes it possible e.g. to add prefix to all table names within a schema, to achieve this enter PREFIX<NAME> in the *target table name* field on the *schema* level.

Schema level:

<b>target schema name</b>	<b>For operations concerning given schema [1] and tables [2] within given schema: schema name will be changed to the name entered here. For selected <i>tables</i> different target schema may be specified.</b>
<b>target table name</b>	For operations concerning tables [2] within given schema: table name will be changed to the name entered here. For obvious reasons the name must contain special sequence <NAME>. For selected <i>tables</i> different target table name may be specified.

Table level:

<b>target schema name</b>	<b>For operations concerning given table [2]: schema name will be changed to the name entered here. This makes it possible to “relocate” selected tables to different schemas.</b>
<b>target table name</b>	For operations concerning given table [2]: table name will be changed to the name entered here.

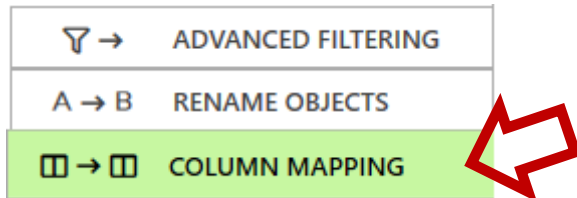
[1] Operations concerning a schema are operations which create/delete/update:

- **View** definition,
- **Code** definition.

[2] Operations concerning a table are operations which create/delete/update:

- **Table** definition,
- **Index** definition,
- **Data**.

#### Column rename and exclude



Use this tab when data from a column in the source database is to be replicated to a column with a different name in the target database, or when a column is to be excluded from replication.

Mapping between source and target columns may be configured manually at the table level. If table name is a template (it contains \* or ? wildcards) it is also required to enter name of the source table (source table name must match the template). To exclude column from replication enter its name and uncheck the **Replicate** checkbox.

MANUAL COLUMN MAPPING				
DELETE X		ADD +		Search Q
Source column name	Replicate	Target column name		
AMOUNT IN EUR	<input checked="" type="checkbox"/>	AMOUNT_IN_EUR		
AMOUNT IN USD	<input checked="" type="checkbox"/>	AMOUNT_IN_USD		
UNUSED_DATA	<input type="checkbox"/>	NOT REPLICATED		

The alternative way of mapping columns is mapping them *automatically*, according to a *rule* specified by the user. Automatic mapping has been introduced in order to facilitate smooth operation of replication in a setup where the maximum column name's length in the target database is less than the maximum column name's length in the source database. When a new column appears with a name the length of which exceeds the maximum allowed in the target database, we want the replicator to automatically trim the name to the required length, which will allow replication to continue. The columns' names which have been generated this way are stored by the replicator and may be viewed on the replication rule screen in the column rename section. If a column name which has been generated automatically is later deemed inappropriate it will be possible to change it at any moment using the following procedure:

1. Stop the replication,
2. Change the column's name in the target database,
3. In configuration at the table level use **CONVERT TO MANUAL** button to move the mapping to the list of manual mappings,
4. Edit target column name in the manual mappings list,
5. Resume the replication.

AUTO COLUMN MAPPING

CHANGE ☐ Limit column names' length Inherited from **database** level.

maximum length 30

Column names will be replaced by shortened names whenever their length exceeds the length specified above. If simply truncating a column name would lead to a duplicate, digits will be appended at the end of the name to obtain a unique name.

Source schema	Source table	Source column name	Target column name
SALES	CUSTOMERS	IS_ELIGIBLE_FOR_SPECIAL_DISCOUNT	IS_ELIGIBLE_FOR_SPECIAL_DISCOU

Automatic mapping may be enabled at any configuration level. Automatic mapping settings will be inherited at the lower levels, unless we choose to alter this behavior using the **CHANGE** checkbox.

Mappings defined in the manual mappings section take precedence over automatically generated ones.

## Database level

DB: Mojabaza

Choose **CUSTOM REPLICATION** if you want to pick schemas and tables to replicate with specific features turned on or off. Choose **EXACT REPLICATION** to obtain a *verbatim copy* of the source database.

☒ configure default **schema** advanced options
 ☒ configure default **table** advanced options

On the database level choose one of the two options: **REPLICATE ALL SCHEMAS AND TABLES EXCEPT SELECTED** or **REPLICATE ONLY SELECTED SCHEMAS AND TABLES**.

**REPLICATE ALL SCHEMAS AND TABLES EXCEPT SELECTED** means that all schemas and tables are replicated, unless they are **excluded** from replication by settings on lower configuration levels. If this option is set then entering default advanced filter settings for schema and table operations is mandatory.

**REPLICATE ONLY SELECTED SCHEMAS AND TABLES** means that only schemas and tables selected for replication on lower configuration levels will be replicated. Entering default advanced filter settings for schema and table operations is optional.

## Schema level

We will need schema level entries whenever we want to:

- **Add** schemas to replicate if **REPLICATE ONLY SELECTED SCHEMAS AND TABLES** is set on the database level,
- **Exclude** schemas from replication if **REPLICATE ALL SCHEMAS AND TABLES EXCEPT SELECTED** is set on the database level,
- Change **settings** for selected schemas.

SCHEMA NAME
+

SCHEMA NAME	TABLE NAME
	+

In order to add schema level configuration entry click on the button with a „plus sign“. “Add schemas to configuration” screen will open:


Enter schema name on the left. You can select one or more schemas from the list, enter schema name (the schema does not have to exist in the database at this point), or enter a **template** containing [wildcards](#).

After schema name is entered, click on one of the buttons available in the middle part of the screen (the actual combination of available buttons depends on which option has been selected on the database level). Schema name will appear on the list on the right.






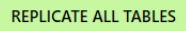
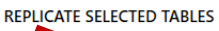
Schema level configuration entries may be of one of the following three types:

<div data-bbox="210 1753 279 1821">+</div> <div data-bbox="331 1771 635 1805">Add schema to replicate.</div>	<div data-bbox="678 1704 1401 1805"> <p><b>Entering advanced filtering options for schema operations is required unless they have been entered on the database level.</b></p> </div> <div data-bbox="678 1809 1401 1877"> <p><b>Table level configuration entries can be entered for tables which belongs to this schema.</b></p> </div>
<div data-bbox="210 1906 279 1928">—</div> <div data-bbox="331 1888 635 1957"> <p><b>Exclude</b> schema from replication.</p> </div>	<div data-bbox="678 1906 1401 1939"> <p>There are no further settings to enter.</p> </div>



 <p>Change <b>settings</b>.</p>	<p>Entering advanced filtering options for schema operations is required unless they have been entered on the database level.</p> <p>Table level configuration entries can be entered for tables which belongs to this schema.</p>
--	--

Next we accept the choice by clicking on **CONFIRM** button. This will cause the schemas we have selected to appear on the main replication rule configuration screen.

SCHEMA NAME	
 <b>FLows_FILES</b>	<div>            Schema FLOWs_FILES will be <b>excluded</b> from replication.         </div>
<div>  <b>TESTDATA</b> </div>	<div> <div>  </div> <div> <div>  </div> <div> <div>  </div> <div>  </div> </div> </div> <div> <input type="checkbox"/> configure this <b>schema</b> advanced options  <input type="checkbox"/> configure default <b>table</b> advanced options         </div> </div>

Entry can be removed by clicking on the button with an “x-mark”.

For entries of type “change settings” and „add to replication” choose one of the two options: **REPLICATE ALL TABLES** or **REPLICATE SELECTED TABLES**.

**REPLICATE ALL TABLES** means that all tables which belong to this schema will be replicated, unless they will are **excluded** on the table configuration level. If this option is set then entering default advanced filter settings for table operations is mandatory.

**REPLICATE SELECTED TABLES** means that only tables selected for replication on table configuration level will be replicated. Entering default advanced filter settings for table operations is optional.

## Table level

We will need table level entries whenever we want to:

- **Add** tables to replication if **REPLICATE SELECTED TABLES** is set on the schema level,
- **Exclude** tables from replication if **REPLICATE ALL SCHEMAS AND TABLES EXCEPT SELECTED** is selected on the database level or **REPLICATE ALL TABLES** is selected on the schema level,
- Change **settings** for selected tables.

SCHEMA NAME	
— FLOWS_FILES	Schema FLOWS_FILES will be <b>excluded</b> from replication.
TESTDATA	<div> <div>REPLICATE ALL TABLES</div> <div>REPLICATE SELECTED TABLES</div> </div> <div> <input type="checkbox"/> configure this <b>schema</b> advanced options  <input type="checkbox"/> configure default <b>table</b> advanced options         </div>
+	

SCHEMA NAME	TABLE NAME
+	

In order to add table level configuration entry click on the button with a „plus sign“. “Add tables to configuration” screen will open:

DBPLUS Data Replicator

DBPLUS Data Replicator

HELP

MAILING

SETTINGS

DISCONNECT

SELECT

IMPORT

Select schema

FIN

Enter table name

☐ LEDGER\_ACCOUNTS  
☐ LEDGER\_ALLOCATIONS  
☐ LEDGER\_BOOKS  
☐ LEDGER\_ENTRIES  
☐ LEDGER\_PARAMETERS  
☐ LEDGER\_PERIODS  
☐ LEDGER\_RULES  
☐ PERIODIC\_REPORTS

REPLICATE >

Add the following tables

+ FIN.LEDGER\_ACCOUNTS

+ FIN.LEDGER\_ALLOCATIONS

+ FIN.LEDGER\_BOOKS

+ FIN.LEDGER\_ENTRIES

to the currently selected

< CANCEL

CONFIRM >

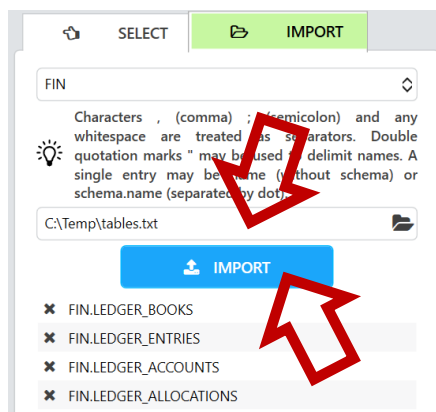
Enter tables to be introduced to configuration on the left. Two methods are available.

## SELECT

Enter schema at the top. You can select from the list of existing schemas, enter schema name (the schema does not have to exist in the database at this point), or enter a template containing [wildcards](#). Below you can select one or more tables from the list (a list appears when a schema which exists in the database has been selected), enter table name (the table does not have to exist in the database at this point), or a **template** containing [wildcards](#).

## IMPORT





In the text box at the top you may enter default schema, it is only necessary if the table set you are about to import contains one or more table names without schema specified. In the text box below enter path to the file you want to import. Clicking on **IMPORT** button will cause table names to be imported and displayed at the bottom. Contents of the imported file are interpreted according to the following rules:

1. Characters: , (comma) ; (semicolon) and whitespace separate entries,
2. Entry may contain table name alone or schema and table name (separated by dot),
3. Double quotes may be used to delimit names.

Examples of valid imported files' content (import of each of these will have the same result provided that FIN has been specified as the default schema):




LEDGER\_BOOKS, LEDGER\_ENTRIES, LEDGER\_ACCOUNTS, LEDGER\_ALLOCATIONS

FIN.LEDGER\_BOOKS;FIN.LEDGER\_ENTRIES;FIN.LEDGER\_ACCOUNTS;FIN.LEDGER\_ALLOCATIONS

"FIN"."LEDGER\_BOOKS" "FIN"."LEDGER\_ENTRIES" "FIN"."LEDGER\_ACCOUNTS"  
"FIN.LEDGER\_ALLOCATIONS"

After entering schema and table names click on one of the buttons available in the middle part of the screen (the actual combination of available buttons depends on which options has been selected on the database and schema level). Table names will appear on the right.

Table level configuration entries may be of one of the following three types:

	<b>Add table to replicate.</b>	<b>Entering advanced filtering options for table operations is required unless they have been entered on the database or schema level.</b>
	<b>Exclude table from replication.</b>	There are no further settings to enter.
	<b>Change settings.</b>	Entering advanced filtering options for table operations is required.

Next we accept the choice by clicking on **CONFIRM** button. This will cause the tables we have selected to appear on the main replication rule configuration screen.

SCHEMA NAME	TABLE NAME	
+ TESTDATA	CUSTOMER	<input type="checkbox"/> configure this <b>table</b> advanced options

Entry can be removed by clicking on the button with an "x-mark".

Button with a „filter“ can be used to change table level configuration entries view mode:

<input checked="" type="radio"/>	<b>Show all tables</b>	<b>All table level entries are displayed.</b>
<input type="radio"/>	<b>Show only tables for selected schema</b>	Only those table level entries are displayed that belong to the schema in the currently selected schema level entry.

### Tablespace mapping

**TABLESPACE MAPPING** tab is available on the **database** level.

☒ → **ADVANCED FILTERING**

☒ **TABLESPACE MAPPING**

Here you can define mapping between tablespaces on source and target databases.

Source tablespace		Target tablespace
* TS_1	USERS	
* TS_2	USERS	

Enter name of a tablespace in the source database in the **Source tablespace** column (name may contain wildcards) and name of a corresponding tablespace in the target database in the **Target tablespace** column. As a result tablespace identifiers will be substituted in DDL commands executed on the target database.

Please note that only tablespaces explicitly specified in DDL commands will be affected, if tablespace has not been specified an object will be created in the default tablespace of the target database.

### Wildcards

In accordance with the well-established convention the following wildcards can be used in the names of schemas and tables:

*	Represents any (possibly empty) sequence of characters.
?	Represents exactly one character.

## Finalizing configuration of replication rules

◀ BACK TO DB SELECTION

TO REPORT ▶

After configuring of replication rules is finished, click on **TO REPORT** button at the lower part of the screen. This will move us to the next stage of the replication creation process.

### Replicated DDL commands depending on source and target databases

Source database	Target database	Replicated DDL commands
ORACLE	ORACLE	In CUSTOM mode: Create, Alter, Drop User Create, Alter, Drop Package/Procedure/Function Create, Alter, Drop View Create, Alter, Drop, Rename Table  In EXACT mode: All DDL commands
SQL Server	SQL Server	Create, Alter, Drop, Rename Table
SQL Server	Oracle	Create, Alter, Drop, Rename Table
ORACLE, SQL Server	PostgreSQL	No DDL commands are replicated
ORACLE	SQL Server	No DDL commands are replicated

### 4.3 Generate and view report

After replication rules are configured system will generate a report. This may take a while. Report contains information about volume of the data selected for replication and potential problems detected by the verification procedure.

DBPLUS Data Replicator

DBPLUS Data Replicator

HELP

MAILING

SETTINGS

DISCONNECT

1. SELECT DATABASES

2. CONFIGURE

3. VIEW REPORT

4. SETUP START OPTIONS

5. NAME & CREATE

SUMMARY

Number of tables 18

Data volume 13.1 MB

Index volume 22.1 MB

SYSTEM CHECK RESULTS

Some of the tables selected for replication have no unique index. This may lead to runtime problems.

FIN.INPUT\_FILES

Archive logs are available

Archive logs are available for the period of 15 days.

◀ BACK TO CONFIGURATION

TO START OPTIONS ▶

Page 28 of 76

You will be able to view the same exact report after the replication creation process is over. Move on to the next stage by clicking on **TO START OPTIONS** button.

## 4.4 Start options

There are two modes of starting up a replication available to choose from: manual and automatic.

### 4.4.1 Manual mode

Starting up replication in manual mode requires to execute the following steps:

1. Set up **start point** in the replication settings.
2. Prepare the target database by populating its tables with data representing state of the source database at the start point.
3. Start replication.

Start point is expressed as number of the last executed database change, it is SCN (System Change Number) in Oracle databases, LSN (Log Sequence Number) In SQL Server databases. The replication process will replicate operations which are part of transactions which **ended after** the moment represented by the start point SCN. In other words a transaction will be replicated if SCN/LSN of its *commit* operation will be **greater than** the SCN/LSN specified as replication start point.

How start point is set up depends on whether the source database is oracle or SQL Server.

Configuring start point differs depending on whether the source database is Oracle or SQL Server.

#### Oracle

**Setup replication start point**

Replication process will transfer transactions committed after the start point. You can use one of the following methods to set up replication start point: 1. Let start point be determined at the moment replication starts as current database SCN. 2. Specify it manually. 3. Let start point be determined by data imported to the target database. The third method will only work when scripts generated by the DBPLUS Data Replicator tools are used for data import.

☐ Current SCN on replication start ☒ Set manually ☐ Automatically set by import

Enter start point

**Transaction log reading start position**

Replicator will automatically calculate the point from which to start reading transaction log to accommodate the start point as specified above. In rare cases you might want to manually override its choice.

☒ Calculate automatically ☐ Set manually

**Data pump script**

☐ Activate data pump script creation tool after this wizard has been finished

#### START POINT [1]

<b>Current SCN on replication start</b>	When replication process starts for the first time value of the current SCN in the source database becomes the replication start point. You can use this option if the source database is not active and will not become active until after the replication has started.
<b>Set manually</b>	<b>Enter</b> the start point SCN number.

<b>Automatically set by import</b>	Start point will be determined based on the last data import performed with the help of the tool for generating <b>data pump</b> data transfer scripts (the tool is part of the DBPLUS Data Replicator product). Script which imports data to the target database saves information about value of the SCN number for which data export has been executed (parameter <i>flashpoint SCN</i> of <i>data pump</i> tool). If option <i>automatically set by import</i> is selected this information will be read during initialization phase of replication process and the flashback SCN of the last export will become the new start point SCN.
------------------------------------	---

## TRANSACTION LOG READING START POSITION [2]

When replication starts for the first time reading from transaction log start at a point *earlier* than the start point, so that all operations which are part of transactions that were active at the moment represented by the start point. If start point is being entered manually you can specify the point at which transaction log reading will start as well.

<b>Calculate automatically</b>	Transaction log reading start position will be calculated <b>automatically</b> to enable replication process to read operations which are part of transactions that were active at the moment represented by the start point. How far the log reading start must be moved back relative to the start point depends on the value of the UNDO_RETENTION parameter.
<b>Set manually</b>	Allows to <b>manually</b> enter value of SCN number at which transaction log reading will start. You can set the transaction log reading start to reduce the time it will take the replication to start for the first time, provided that it is known that there are no earlier operations which need to be replicated.

Setting the option within the **Data pump script [3]** section will cause data pump script generation tool to open after replication creation has been finished. [How to use the tool is explained here.](#)

## SQL Server

For SQL Server source database start point can be enter only after replication has been created. This is because change capture process in the source database starts at the moment replication is created (change capture runs continuously in the background, whether the replication process itself is active or not). An LSN value from *before* the moment replication starts does not constitute a valid replication start point.

After replication has been created but before it is started for the first time, set up the start point at the replication settings. To do this press the **CONFIG** button and then select the **START & END** tab.

### Setup replication start point

Replication process will transfer transactions committed after the start point. You can use one of the following methods to set up replication start point: 1. Let start point be determined at the moment replication starts as current database LSN. 2. Specify it manually.

☐ Current LSN on replication start
 ☒ Set manually


<b>Current LSN on replication start</b>	When replication process starts for the first time value of the current LSN in the source database becomes the replication start point. You can use this option if the source database is not active and will not become active until after the replication has started.
<b>Set manually</b>	<b>Enter</b> the start point LSN number.

#### 4.4.2 Automatic mode

In automatic mode replicator will initialize the replication by sending and loading the initial data on the target database for all replicated tables.

To take advantage of this mode DBPLUS Data Transfer tool must be available. Configure Data Transfer service connection settings by providing host and port number. The default port number is 51220.

**Data Transfer service**

 In order to use automatic start mode DBPLUS Data Transfer service must be set up. DBPLUS Data Transfer service is used to transfer initial data for replicated tables between source and target databases.

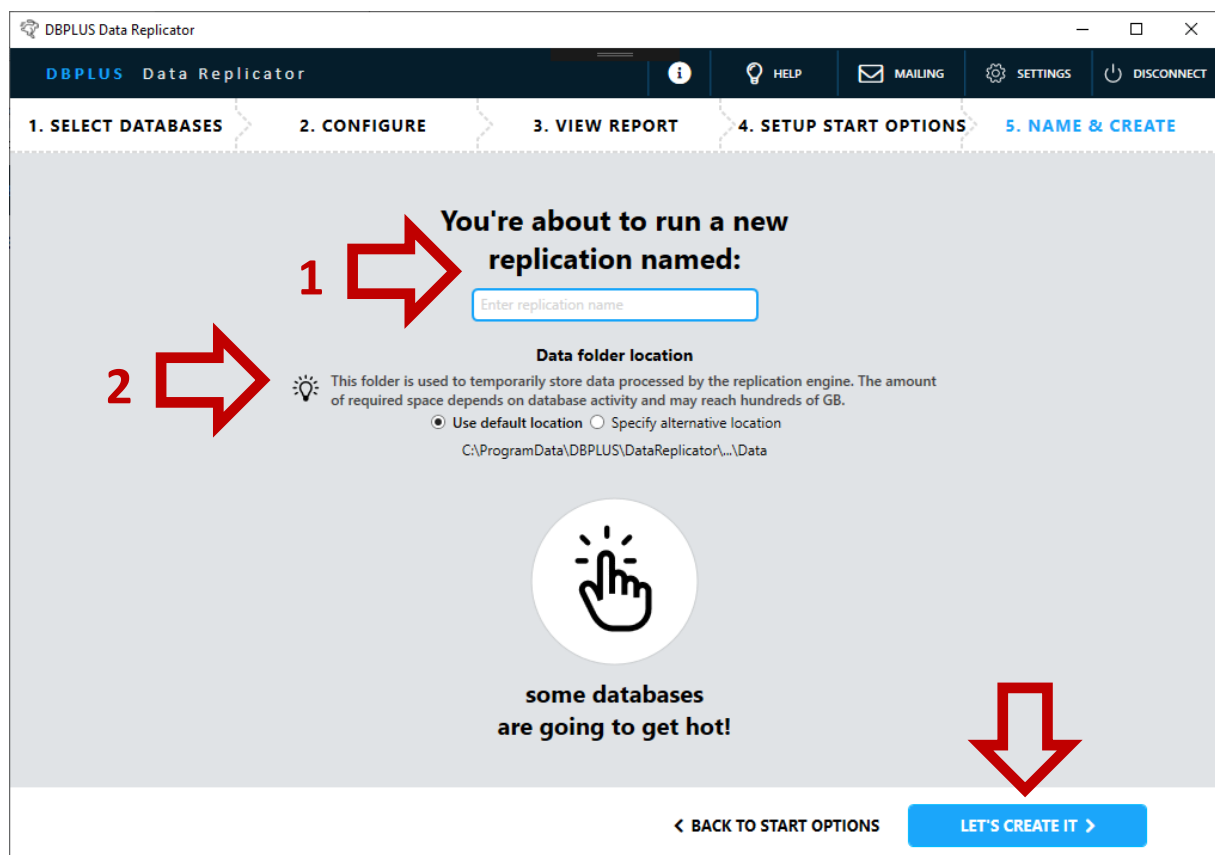
server

localhost

port

(Use default)

#### 4.5 Name and create the replication




DBPLUS Data Replicator



**DBPLUS Data Replicator** | [HELP](#) | [MAILING](#) | [SETTINGS](#) | [DISCONNECT](#)

1. SELECT DATABASES | 2. CONFIGURE | 3. VIEW REPORT | 4. SETUP START OPTIONS | **5. NAME & CREATE**

**You're about to run a new replication named:**


1  Enter replication name

**Data folder location**


2   This folder is used to temporarily store data processed by the replication engine. The amount of required space depends on database activity and may reach hundreds of GB.

☒ Use default location ☐ Specify alternative location

C:\ProgramData\DBPLUS\DataReplicator\...\Data



**some databases are going to get hot!**



[< BACK TO START OPTIONS](#) [LET'S CREATE IT >](#)

There are still a few settings left before we can finish.



### NAME [1]

Name may contain only alphanumeric characters (letters and digits) and underscore (character \_).

### DATA FOLDER LOCATION [2]

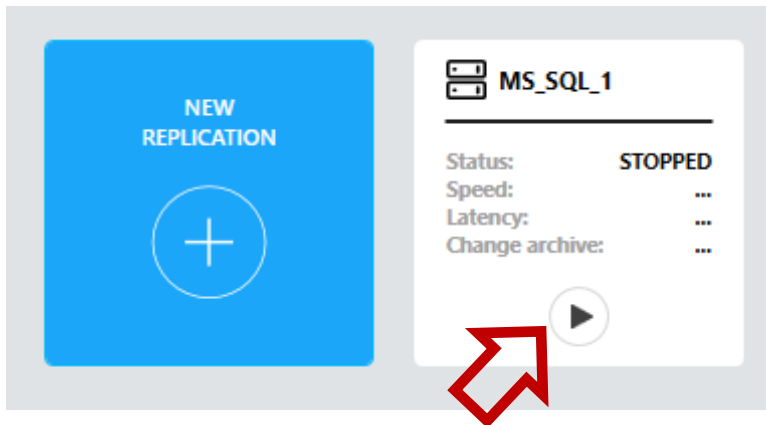
By default the folder is located on disk C, the path is:

C:\ProgramData\DBPLUS\DataReplicator\(replication name)\Data

If locating data folder on the C disk is not ideal choose **Specify alternative location** option and enter path for the data folder. Remember that the folder must be accessible from the machine where the replication service runs.

After completing the setup, click on **LET'S CREATE IT** button. You will be transferred to the main screen where the newly created replication will appear. The new replication is *not yet started*. You have to start it – we will talk about this in the [next chapter](#).

## 5 Starting replication



After new replication is configured and created it appears on the panel of replication cards. Before starting it you can verify and change its [advanced settings](#). To start the replication click on the “start” button on the replication card.

### 5.1 Prerequisites

In order for the replication process to work a number of preconditions must be satisfied. For the convenience of the user all prerequisites are automatically verified during the starting phase of the replication process. If any of the prerequisites is found missing an [alert](#) is raised.

[List of prerequisites for Oracle databases](#)

#### Source database

- Privilege **SELECT ANY TRANSACTION**
- Role **EXECUTE\_CATALOG\_ROLE**
- Privilege **SELECT ANY DICTIONARY** or role **SELECT\_CATALOG\_ROLE**
- **ARCHIVE LOG** mode enabled
- **SUPPLEMENTAL LOG** enabled with options **PRIMARY KEY COLUMNS** and **UNIQUE COLUMNS**
- Package **dbms\_stats** is installed and accessible for the user on which the replication runs.

#### Target database

- Privilege **SELECT ANY DICTIONARY** or role **SELECT\_CATALOG\_ROLE**
- Permissions required to create schema (user) and add tables, indexes and packages to it.
- Permissions required to execute the replicated operations.

[List of prerequisites for SQL Server databases](#)

#### Target database

- Permission **VIEW DEFINITION**
- Permissions required to create schema (user) and add tables and indexes to it.
- Permissions required to execute the replicated operations.

[List of prerequisites for PostgreSQL databases](#)

#### Target database

- Permission **SELECT** on the following views: information\_schema.columns, pg\_catalog.pg\_database, pg\_catalog.pg\_user, pg\_catalog.pg\_index, pg\_catalog.pg\_namespace, pg\_catalog.pg\_class
- Permissions required to create schema (user) and add tables and indexes to it.
- Permissions required to execute the replicated operations.

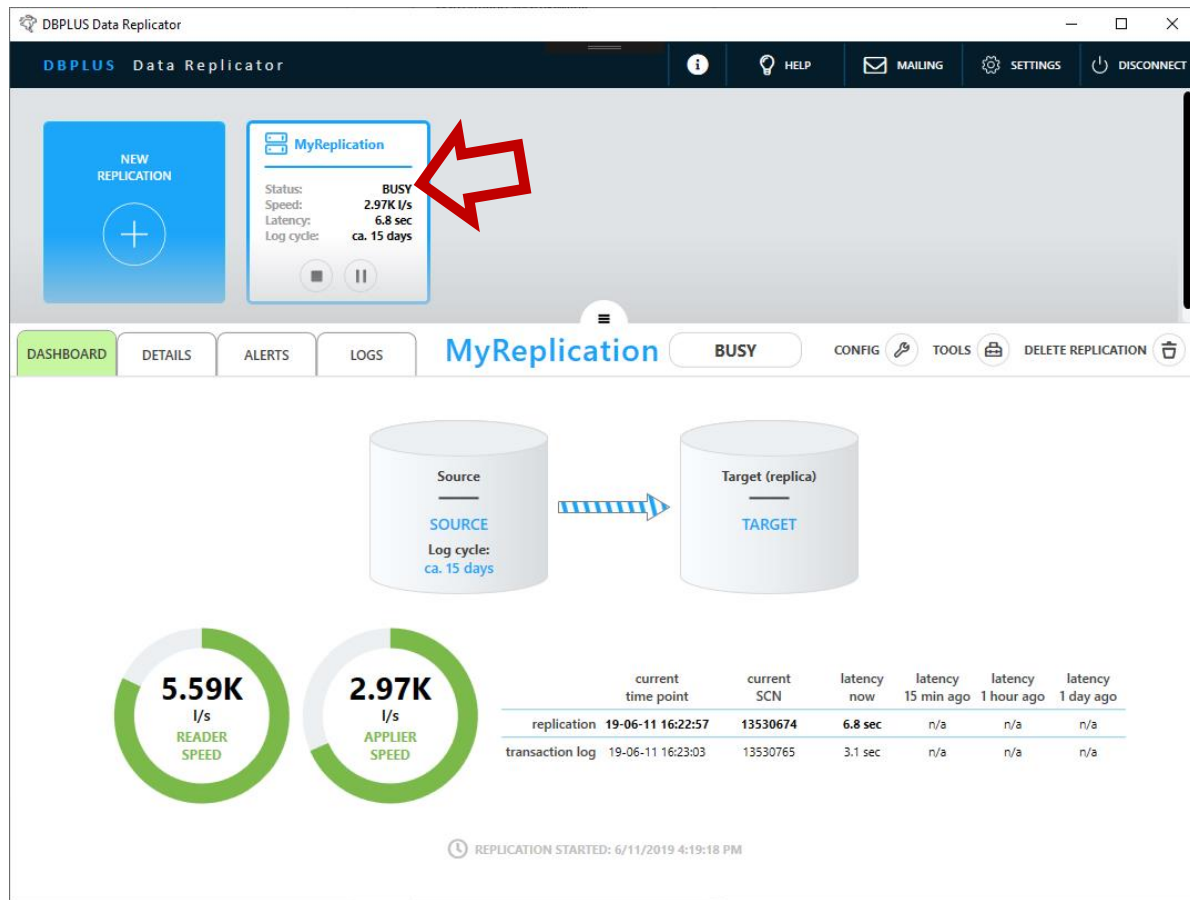
## 5.2 Artefacts

Replicator does not create any artefacts in the source database.

In the target database replicator creates a schema (user) named DBPLUSDR\_{replication name} (the name of the schema can be changed with the help of [advanced settings](#)). In this schema replicator creates all objects used by the replication process.

## 6 Replication monitoring and management

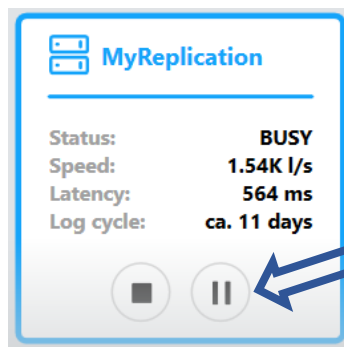
After selecting a replication from the available replication cards the management panel will appear at the bottom part of the screen.



### 6.1 Replication states

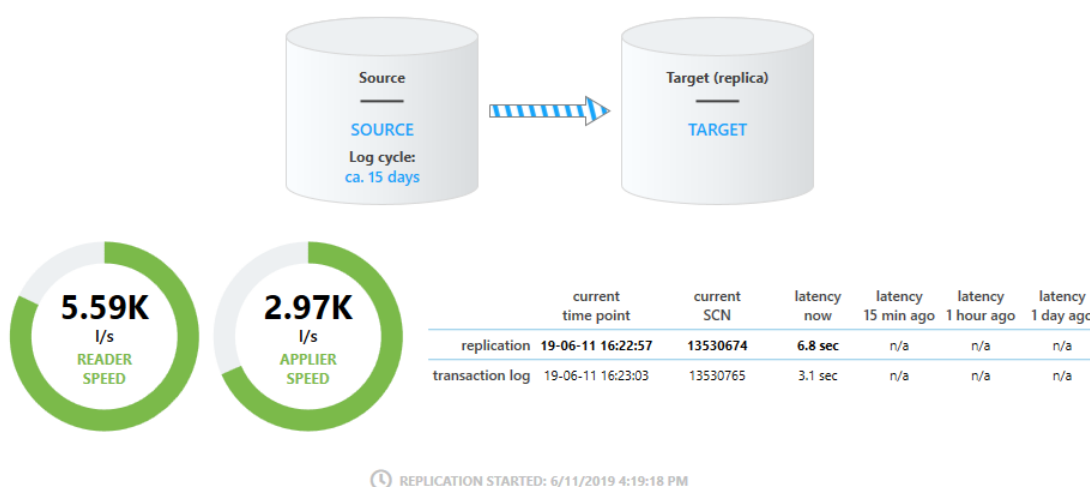
At any time replication process can be in any of the following states:

BUSY	Replication process is on and working.
IDLE	Replication process is on but currently there is nothing for it to do.
ERROR	Replication process is on, but is not working (or will soon cease to work) because of an error. If replication process is in this state there will be at least one active alert for this replication with severity level "high".
PAUSED	Replication process is on but it has been paused. When in this state replication process uses system resources such as RAM and database connections.
STOPPED	Replication process is off.
FINISHED	Replication has reached its <i>end point</i> . Replication process is on but is not working.
Intermediate states	
STARTING	State is changing to STARTED.
PAUSING	State is changing to PAUSED.
STOPPING	State is changing to STOPPED.
DELETING	Replication is currently being deleted.



Changing replication's state is achieved by clicking on the buttons at the lower part of a replication card.

## 6.2 Statistics



<b>READER SPEED</b>	<p>Transaction log reading speed expressed as number of transaction log lines per second.</p> <p>Fill and color of the strip is based on comparison of the current (momentary) reading speed and the average speed at which new lines are “generated” by the source database. Green color means we are reading lines at least with the same speed as the source database is generating them (we’re keeping up); red color means we are reading considerably slower than the source database is generating them (we’re falling behind).</p>
<b>APPLIER SPEED</b>	<p>Speed of applying changes in the target database expressed as number of transaction log lines, which constitute the source of the operations being applied, per second. Applier speed may be lower than reading speed because not all operations which are read from transaction log end up being applied in the target database.</p> <p>Fill and color of the strip is based on comparison of the current (momentary) applier speed and average speed at which new applicable operations appear in the source database. Green color means we are applying operations at least with the same speed as the source database is generating them (we’re keeping up); red color means we are applying operations considerably slower than the source database is generating them (we’re falling behind).</p>
<b>Log cycle</b>	Estimation of for how long transaction logs are retained in the source database.

replication

<b>current time point</b>	Time point in the source database which the current state of the target database represents, in other words the original time of execution of the last change applied in the target database.
<b>current SCN/LSN</b>	System change number of the source database which the current state of the target database represents, in other words the system change number (in the source database) of the last change applied in the target database.
<b>latency now</b>	<p>Current latency between source and target database.</p> <p>Two methods of calculating latency are being used:</p> <ol style="list-style-type: none"> <li>1. As difference between the current time in the source database and the latest time point which the data in the target database represents.</li> <li>2. As difference between the time of execution in the target database and the time of execution in the source database of the last applied change.</li> </ol> <p>The latest time point which the data in the target database represents is the time point which <i>precedes</i> the next change which awaits execution. If there are no changes awaiting execution in the replicator's pipeline we assume (conservatively) that a next change to be replicated will arrive immediately.</p> <p>The primary way of calculating latency is method (1). When replicator is applying changes in the target database (replication is active) we use method (2) in addition to method (1), calculating latency as <i>maximum</i> of the results arrived at by using both methods. This is to avoid a situation when the reported latency drops abruptly when the last bundle of changes has been applied and then grows steadily while the next bundle of changes is being processed.</p> <p>If replication is IDLE, symbol – (hyphen) is displayed.</p>
<b>latency ... ago</b>	Historical values of replication latency. If appropriate historical entry cannot be found text n/a (not available) is displayed.
<b>transaction log</b>	
<b>current time point</b>	Write time (to the transaction log) of the last line read from transaction log in the source database.
<b>current SCN/LSN</b>	System change number of the last line read from transaction log in the source database.
<b>latency now</b>	<p>Current latency of transaction log reading process.</p> <p>Latency is calculated as difference between the current time and the write time (to the transaction log) of the last line read from transaction log in the source database.</p> <p>If the log reading process has reached the end of the log (there is no more data to read), symbol – (hyphen) is displayed.</p>
<b>latency ... ago</b>	Historical values of transaction log reading latency. If appropriate historical entry cannot be found text n/a (not available) is displayed.

### 6.3 Table level monitoring and management

To access table level monitoring and management select **DETAILS** tab on the replication panel.

The screenshot shows the DBPLUS Data Replicator web interface. At the top, there's a navigation bar with 'DBPLUS Data Replicator', 'HELP', 'MAILING', 'SETTINGS', and 'DISCONNECT'. Below this, a 'NEW REPLICATION' button is on the left, and a 'MyReplication' status box shows 'Status: IDLE', 'Speed: ...', 'Latency: ...', and 'Log cycle: ca. 15 days'. The main content area has tabs for 'DASHBOARD', 'DETAILS' (selected), 'ALERTS', and 'LOGS'. Below the tabs, there's a 'Replicated tables' section with a 'SHOW HISTORY' button, a 'SHOW WARNINGS' button, and a 'SHOW' dropdown menu set to 'only currently active'. A search bar is also present. The main table lists replicated tables with columns: Schema, Name, Operations, Last SCN, Last timestamp, and Status. The table contains 15 rows of data, including tables like CUSTOMERS, ORDERS, ORDER\_LINES, etc. The 'TAX\_CODES' row is highlighted, showing a status of 'REPLICATING' and a progress bar for 'INITIAL LOAD' at 18%.

Schema	Name	Operations	Last SCN	Last timestamp	Status
FIN	CUSTOMERS	1036	13533169	2019-06-11 16:38:23	REPLICATING
FIN	ORDERS	1074	13541130	2019-06-11 17:14:19	REPLICATING
FIN	ORDER_LINES	3456	13541215	2019-06-11 17:14:54	REPLICATING
FIN	ORDER_COMMENTS	55	13541296	2019-06-11 17:15:15	REPLICATING
FIN	ORDER_TAX	471	13541345	2019-06-11 17:15:32	REPLICATING
FIN	LANGUAGES	2	13541455	2019-06-11 17:15:43	REPLICATING
FIN	ADDRESSES	1425	13541526	2019-06-11 17:16:07	REPLICATING
FIN	ORDER_HISTORY	8966	13541866	2019-06-11 17:16:25	INITIAL LOAD
FIN	TAX_CODES	39	13541915	2019-06-11 17:16:43	REPLICATING
FIN	STORES	52	13542514	2019-06-11 17:19:51	REPLICATING
FIN	LOCATIONS	8	13542584	2019-06-11 17:20:04	REPLICATING
FIN	PAYMENT_METHODS	90	13542466	2019-06-11 17:19:26	REPLICATING
FIN	PRODUCT_LOCATION!	119	13542429	2019-06-11 17:19:13	REPLICATING
FIN	PRODUCT_PRICES	1010	13542383	2019-06-11 17:18:57	REPLICATING

Grid on the DETAILS tab shows all tables which are currently eligible for replication. There are two types of events which result in a table being added to this list: modifying replication rules in a way which expands the set of tables meeting the criteria, or creating a new table in the source database (execution of a CREATE TABLE command) which meets the replication rules criteria. Similarly there are two types of event which result in a table being removed from this list: modifying replication rules in a way which shrinks the set of tables meeting the criteria, or dropping a table (execution of a DROP TABLE command). In the latter case the table is removed from the list only after it has been dropped on the target database.

Historical information about tables which once were but no longer are eligible for replication can be accessed by selecting option **all (including removed or dropped)** in the **SHOW** combo box.

All possible states of a table eligible for replication and actions accessible in those states are listed in the table below.

## REPLICATING

Table is being replicated. Data in this table in the target database is a replica of the data in its counterpart in the source database (at the point in time which corresponds to the replication progress).

### REINITIALIZE

Table will be reinitialized for replication. Data currently held by this table in the target database will be deleted, then data in its counterpart on the source database will be sent and loaded onto the target database.

### SET START POINT

	Replicating of this table's data will be disabled until the specified point (table's start point) has been reached by the replication.
<b>INITIAL LOAD</b>	Initial load process is in progress. State of this table's data in the target database is undefined. Table in the target database cannot be used.
<i>SCHEDULED</i>	Awaiting for the initial load to commence.
	You can <b>pause</b> , <b>resume</b> or <b>cancel</b> the initial load process.
<i>TRANSFERRING DATA</i>	Initial data for this table is being transferred and loaded onto the target database by the Data Transfer tool.
	You can <b>pause</b> [*], <b>resume</b> [*] or <b>cancel</b> [*] the initial load process.
<i>FINALIZING</i>	At this point initial data transfer by the Data Transfer tool has been completed. To close the initial load process we still need to: <ul style="list-style-type: none"> <li>• Wait for the replication to apply changes up to the time point when reading of the table data by the Transfer Tool has ended,</li> <li>• Merge changes which occurred in this table in the source database in the meantime.</li> </ul>
	You can <b>cancel</b> the initial load process.
<i>ERROR/CANCELED</i>	Initial load process has ended with an error/was canceled by the user.
	<b>RETRY</b> Initial load process will be restarted.
	<b>START REPLICATING</b> Skip initial load and start replicating data for this table <i>at once</i> or <i>at the specified point</i> (table's start point).
<b>START POINT SCHEDULED</b>	Changes for this table will not be replicated until replication has reached the specified point (table's start point). Once this point has been reached table's state will be changed to REPLICATING.
	<b>CHANGE</b> You can change the <i>start point</i> or <i>remove it</i> and change table's state to REPLICATING.

[\*] This action requires that the replicator service is running.

To execute the same operation for multiple tables in bulk select appropriate rows on the grid, then a panel will appear at the bottom of the panel containing buttons for actions which can be executed for all the selected tables.

Here are a few important facts about the automatic initial load feature:

- Automatic initial load feature utilizes an external tool DBPLUS Data Transfer. It has to be configured on the [basic settings configuration page](#).
- Automatic initial load feature does not take *foreign keys* into account. If your data model contains foreign keys you have to remove foreign keys leading from and to the tables which you plan to be automatically initialized for replication, and then restore them once initialization is complete.
- For tables in INITIAL LOAD TRANSFERRING DATA state data transfer and loading will be executed by Data Transfer tool (unless it has been stopped) even when the replicator service is stopped.

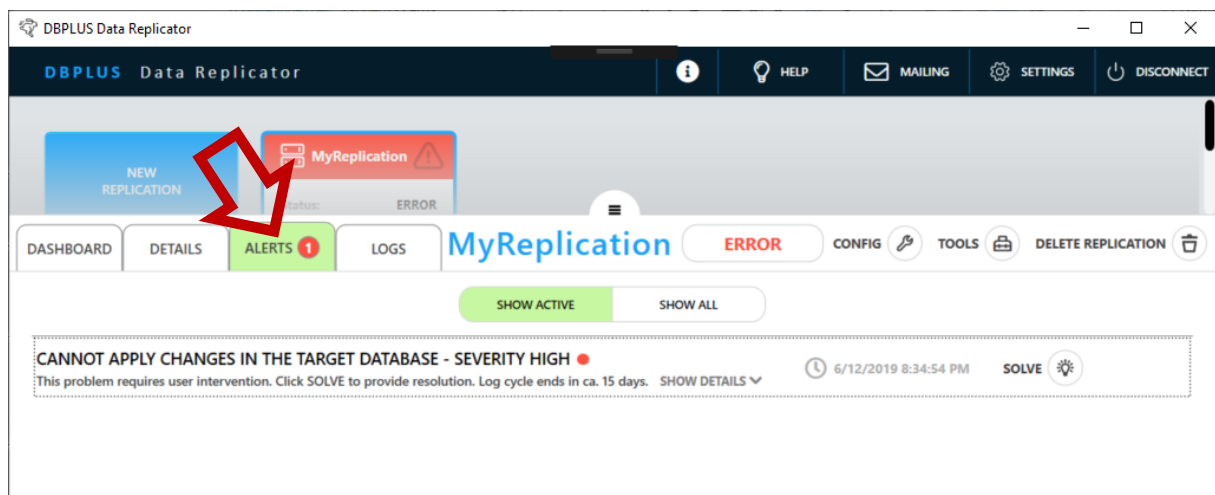


- If applier is brought to a halt (e.g. because of an applier error) it makes it impossible to start transferring initial data and in effect tables will get stuck in INITIAL LOAD SCHEDULED state.

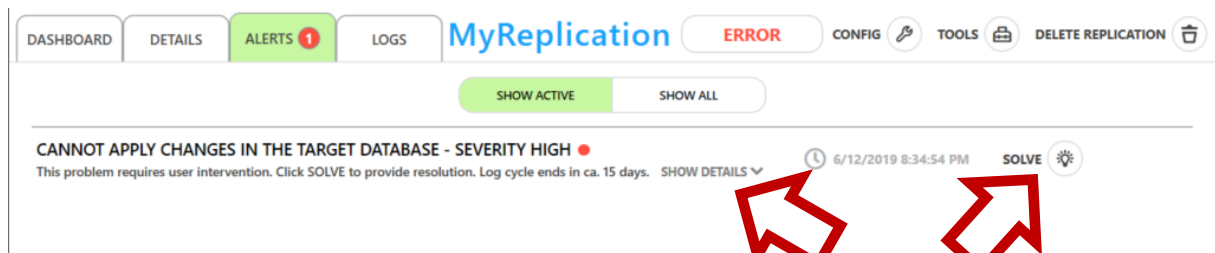
## 6.4 Alerts

Alerts provide notifications about situations which require user intervention. Some alerts are associated with tools which help to solve particular problems.

An active alert is signaled by the red color of the replication card and appearance of a red circle with number of active alerts on the **ALERTS** tab. To open the alert list click on the tab.

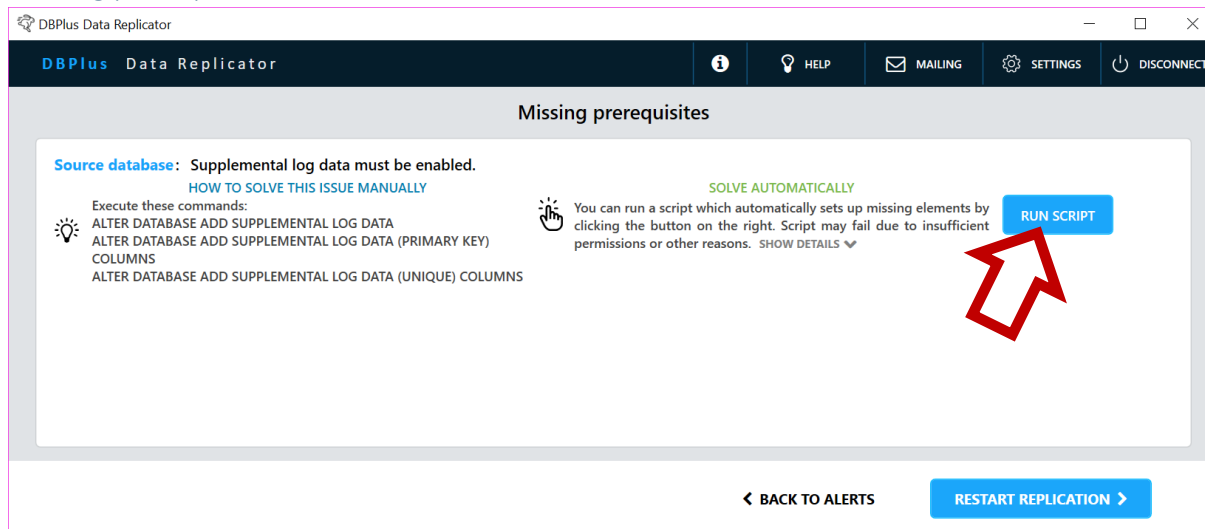


After you click the tab, panel with list of alerts will appear. To get more information about an alert click on **SHOW DETAILS**. If a tool is available which will assist you with solving the problem there will be a button with text **SOLVE** on the right. Click on it to open the associated problem assistance tool.



The program provides two tools for solving problems: for missing prerequisites and for errors encountered when applying changes in the target database.

## Missing prerequisites



DBPLUS Data Replicator

Missing prerequisites

**Source database:** Supplemental log data must be enabled.

**HOW TO SOLVE THIS ISSUE MANUALLY**

Execute these commands:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY)
COLUMNS
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (UNIQUE) COLUMNS
```

**SOLVE AUTOMATICALLY**

You can run a script which automatically sets up missing elements by clicking the button on the right. Script may fail due to insufficient permissions or other reasons. [SHOW DETAILS](#)

**RUN SCRIPT**

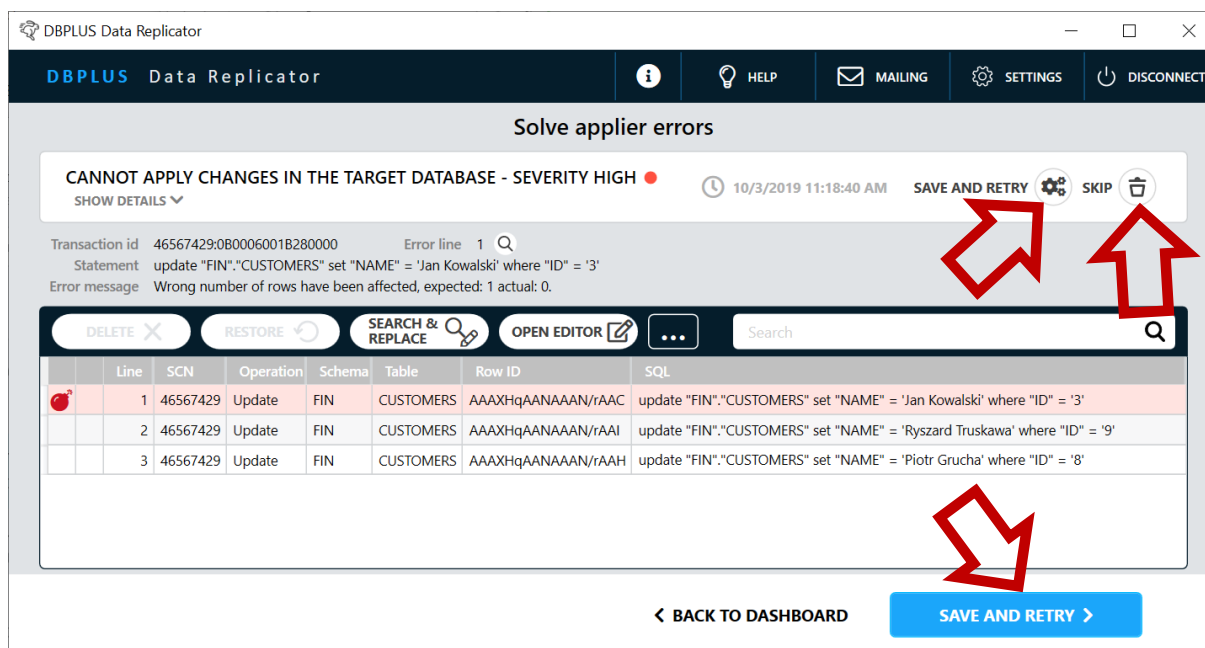
[BACK TO ALERTS](#) [RESTART REPLICATION](#)

On the missing prerequisites panel a list is displayed which contains an entry for each of the missing prerequisites detected. If given missing prerequisite can be handled by executing a known sequence of actions a block named **HOW TO SOLVE THIS ISSUE MANUALLY** will appear. If in addition there is a possibility to automatically take care of the problem by executing a built-in script a block named **SOLVE AUTOMATICALLY** and **RUN SCRIPT** button will also be displayed.

After problems are taken care of (manually or automatically) restart of replication process is required.

## Errors encountered when applying changes in the target database (applier errors)

This kind of situation occurs when an operation (which is an equivalent of an operation previously executed in the source database) fails to execute properly in the target database. Reasons may vary, the simplest cause for such a problem is an earlier change of the target database performed by another program or directly by a user.



DBPLUS Data Replicator

Solve applier errors

**CANNOT APPLY CHANGES IN THE TARGET DATABASE - SEVERITY HIGH**

[SHOW DETAILS](#) 10/3/2019 11:18:40 AM [SAVE AND RETRY](#) [SKIP](#)

Transaction id 46567429:0B0006001B280000 Error line 1

Statement update "FIN"."CUSTOMERS" set "NAME" = 'Jan Kowalski' where "ID" = '3'

Error message Wrong number of rows have been affected, expected: 1 actual: 0.

Line	SCN	Operation	Schema	Table	Row ID	SQL
1	46567429	Update	FIN	CUSTOMERS	AAAXHqAANAAAN/rAAC	update "FIN"."CUSTOMERS" set "NAME" = 'Jan Kowalski' where "ID" = '3'
2	46567429	Update	FIN	CUSTOMERS	AAAXHqAANAAAN/rAAI	update "FIN"."CUSTOMERS" set "NAME" = 'Ryszard Truskawa' where "ID" = '9'
3	46567429	Update	FIN	CUSTOMERS	AAAXHqAANAAAN/rAAH	update "FIN"."CUSTOMERS" set "NAME" = 'Piotr Grucha' where "ID" = '8'

[DELETE](#) [RESTORE](#) [SEARCH & REPLACE](#) [OPEN EDITOR](#) [...](#)

[BACK TO DASHBOARD](#) [SAVE AND RETRY](#)

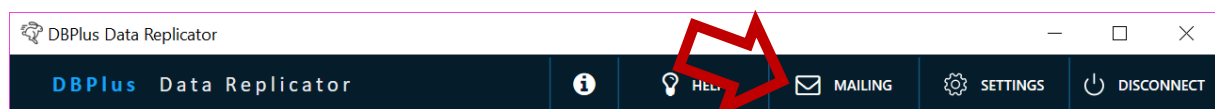
A complete list of operations (SQL statements or PL/SQL scripts) of the transaction which failed to execute will appear on the screen. The line where the error occurred will be colored red and marked with the symbol of a “bomb”.

You may edit text of any operations or delete them using **DELETE** button. At any moment you can restore original content of any previously edited or deleted lines using **RESTORE** button. The **DELETE** and **RESTORE** commands work on **all the selected** lines. After editing is completed, click on **SAVE AND RETRY** button to send changes to the replication process and attempt to execute the transaction in corrected form. You can also use **SKIP** command, which causes replication process to skip the entire transaction which caused problem and continue with the replication.

The following tools are available to help with editing the operations:

<b>SEARCH &amp; REPLACE</b>	Searches for a string of characters in all selected lines and replaces it with another.
<b>TRIM TO KEY</b>	For each selected UPDATE or DELETE operation trims the WHERE clause by retaining only conditions pertaining to key columns and removing all others. Key columns are columns contained in: <ul style="list-style-type: none"> <li>• The primary key, if table has primary key,</li> <li>• The shortest unique key (only unique keys which in fact guarantee uniqueness of column values), if table has no primary key,</li> <li>• If there are no unique keys the WHERE clause will not be modified.</li> </ul>
<b>TRANSFORM TO LITERALS</b>	For each selected DML operation (INSERT, UPDATE, DELETE) with bind variables modifies the SQL statement by replacing all references to bind variables with literals. The parameters are not removed.

## 6.5 Automatic email notifications concerning replication problems



You can set up the system to automatically send email notifications whenever replication problems (alerts) appear. Configuration of email notifications is done globally (once for all replications), one can however specify which recipients are interested in notifications concerning which replications.

**1**

To turn automatic email notifications on check **send alert notifications by email** [1]. Then enter SMTP client setup [2]. We will need a user authorized to send email messages.

<b>host</b>	Name or IP address of the SMTP server.
<b>port</b>	Port number of the SMTP service. Defaults to 25 (no SSL) or 465 (with SSL).
<b>enable SSL</b>	Whether to use secure connection with SSL?
<b>user email</b>	Email address of the user, whose account will be used by the email notification sending feature.
<b>user name</b>	Name of the user, whose account will be used by the email notification sending feature. User name and password may be left empty if you use a (test) server that does not require authentication.
<b>password</b>	User's password.

To check the SMTP client setup send test email message using the **SEND TEST MESSAGE** button [3].

Other options are entered in section [4].

<b>Choose alerts to send</b>	Specify minimum severity level required for an alert to trigger an email notification.
<b>Resend alerts if not closed</b>	Whether and how often to resend notifications about alerts which remain open. A notification concerning each alert is sent to each recipient at least once.
<b>do not resend</b>	Never send notification about the same alert to the same recipient more than once.
<b>daily</b>	Send notification again if alert remains open and at least 24 hours have passed since the notification was last sent.
<b>weekly</b>	Send notification again if alert remains open and at least 7 days have passed since the notification was last sent.

It is necessary to enter recipients [5]. To each recipient a list of replications he or she is interested in may be assigned. By default a recipient is assumed to be interested in all replications.

Save all settings using the **ACCEPT CHANGES** button [6].

A few facts concerning the notification generation mechanism:

1. The mechanism is run once each 15 minutes **and** immediately after each configuration change.
2. Each time at most one message is sent to one recipient, if there is more than one alert which triggered a notification a summary message is generated.
3. If more than one recipient is to receive a message with exactly the same content one message will be sent with all the recipients listed in the "send to" field.
4. In case of a message delivery failure attempt will be repeated each 15 minutes.

On the **history & errors** tab one can browse through a list of recently sent messages and check for active error messages.

The screenshot shows the 'History & errors' tab. It contains two main sections: 'Recent messages' and 'Mailing errors'. Both sections have a search bar. The 'Recent messages' section displays a table with the following data:

Send date/time	Delivered to	Subject
10/27/2017 5:47:56 PM	john.smith@megacorp.org, rachel.abraham@megac...	2 issues in replication BI

The 'Mailing errors' section is currently empty. A 'CLOSE' button is located at the bottom of the panel.

The **Mailing error** list presents only *active errors*, that is errors which are currently preventing delivery of messages with notifications of replication problems.

## 6.6 Configuration edit

The screenshot shows the 'MyReplication' management panel. The navigation bar includes buttons for 'DASHBOARD', 'DETAILS', 'ALERTS', 'LOGS', 'MyReplication', 'BUSY', 'CONFIG', 'TOOLS', and 'DELETE REPLICATION'. A red arrow points to the 'CONFIG' button, which is used to access the configuration edit screen.

To view or edit replication configuration or use any of the maintenance tools use **CONFIG** button on the replication management panel. Configuration edit screen will appear.

The screenshot shows the configuration edit screen with eight tabs: 'RULES', 'CONNECTION', 'START & END', 'BASIC SETTINGS', 'ADVANCED', 'CONVERSION', 'ALERTS', and 'SCHEDULER'. The 'RULES' tab is currently selected.

The screen consists of eight tabs:

**RULES** provides configuration rules [described earlier](#).

**CONNECTION** provides configuration of database connections [described earlier](#).

**START & EDIT** provides [start point and end point settings](#).

**BASIC SETTINGS** provides [basic configuration options](#).

**ADVANCED** provides [advanced configuration options](#).

**CONVERSION** provides [data conversion and data type mapping rules](#).


**ALERTS** provides [configuration of optional alerts](#).

**SCHEDULER** provides [planned replicator breaks schedule](#).

## Start point and end point settings

Start point options can be modified as long as the replication has not yet been started for the first time. Setting up start point change is described in a [separate chapter](#).

**End**


 Specify end point SCN if you want the replication to stop at some point. Replication process will transfer transactions committed at or before the end point.

☒ Run forever (no end point)
 ☐ Specify end point

End point allows to stop replication after replicating changes up to a specified point. Replicator will apply transactions with *commit* change number **less or equal** the end point entered here.

## Basic configuration

**Data folder location**

 This folder is used to temporarily store data processed by the replication engine. The amount of required space depends on database activity and may reach hundreds of GB.

☒ Use default location
 ☐ Specify alternative location


C:\ProgramData\DBPLUS\DataReplicator\MyReplication\Data

By default the folder is located on disk C, the path is:

C:\ProgramData\DBPLUS\DataReplicator\{(replication name)}\Data

If locating data folder on the C disk is not ideal choose **Specify alternative location** option and enter path for the data folder. Remember that the folder must be accessible from the machine where the replication service runs.

**Service start mode**


 Select startup mode for the Windows service dedicated to run this replication. Set to "Automatic" if you want the replication process to start automatically when Windows starts.

Automatic

⬆

Here you can change *start mode* for the Windows service the replicator runs on. This is the same option which you can view or edit in Windows Task Manager program.

**Data Transfer service**

 DBPLUS Data Transfer service is used to transfer initial data for replicated tables between source and target databases.

server

port

password

**Options**

☒ **Configure** ☐ Use default settings

when target table already exists

☒ **truncate it** ☐ drop and recreate it

☒ and copy

☐ only primary indexes

☒ **all unique**

☐ all

☒ **UNIQUE** constraints

when creating target table create

☐ only primary indexes

☐ all unique indexes

☒ **all**

☒ **NOT NULL** constraints

☒ **UNIQUE** constraints

maximum concurrent tasks

maximum attempts

retry wait time  d  h  m

issue warnings when data is truncated to fit target column ☐

data read method ☒ **single select** ☐ partitioned

compression ☐ disabled ☒ **enabled**


This section provides setup required to use DBPLUS Data Transfer tool, which is utilized by the automatic initial load feature.

Data Transfer service	
<b>server</b>	Machine where the DBPLUS Data Transfer service is running.
<b>port</b>	Port the Data Transfer service is listening on, the default port number is 51220.
<b>password</b>	Password to the Data Transfer service (empty, if password has not been set up on the service side).
Options	
<b>when target table already exists</b>	<p>When table being initialized already exists in the target database...</p> <p><b>truncate it</b> use the existing table, all data from the target table will be erased before data transfer is started.</p> <p><b>recreate it</b> drop it and then create it on the basis of the source table. When using existing table it is possible to copy indexes from the source table to the target table. All indexes that already exist on the target table will be preserved. It is possible to copy ...</p> <p><b>only primary index</b> only primary key index.</p>

	<p>If the source table has unique indexes but no primary key one unique key will be created.</p> <p><b>all unique indexes</b> all unique indexes.</p> <p><b>all indexes</b> all indexes.</p> <p>If <b>UNIQUE</b> option is enabled unique indexes associated with unique constraints will be copied along with constraints, otherwise only indexes will be copied.</p>
<b>when creating table create</b>	<p>When creating table in the target database on the basis of a table from source database create indexes ...</p> <p><b>only primary index</b> only primary key index. If the source table has unique indexes but no primary key one unique key will be created.</p> <p><b>all unique indexes</b> all unique indexes.</p> <p><b>all indexes</b> all indexes.</p> <p>And constraints:</p> <p><b>NOT NULL</b> NOT NULL constraints on table columns. This option works in conjunction with creation of unique indexes. If enabled unique indexes associated with unique constraints will be copied along with constraints, otherwise only indexes will be created.</p> <p><b>UNIQUE</b></p>
<b>maximum concurrent tasks</b>	Maximum number of data transfer task being executed simultaneously.
<b>maximum attempts</b>	Maximum number of times an initial data transfer is to be attempted, applicable when an attempt to transfer data fails. Value of 1 means that only one attempt will be made, in other words there will be no attempts to retry after Data Transfer tool reports an error.
<b>retry wait time</b>	Time to wait when an initial data transfer task fails with an error before next attempt can be made.
<b>issue warnings when data is truncated to fit target column</b>	If this option is enabled, whenever truncation or rounding is necessary to fit data into a column in the target table a warning will be issued. In order for conversions leading to data loss to be performed appropriate options in <a href="#">data conversion rules</a> section must be enabled, otherwise inability to write data into the target table without data loss will result in <i>initial data load being aborted with an error</i> .
<b>data read method</b>	<p>How data will be read from the source table</p> <p><b>single select</b> A single query will be issued on the source table to read all rows.</p> <p><b>partitioned</b> The source table will be analyzed and divided into partitions, then rows from each partition will be read using a separate query.</p>
<b>compression</b>	Whether to enable compression for data sent over the network from the reader (reading from the source database) to the writer (writing to the target database). Compression reduces volume of data sent over the network, but increases CPU usage.



**Metadata columns**

 Enable metadata columns to get information about last update on each record in the target database.

☒ source change number  
☒ source timestamp  
☒ source transaction id

drop metadata columns that are no longer enabled ☐

Metadata columns are added to tables in the target database and provide information about the last update made to any given record.

<b>source change number</b>	<p>Enables DBPLUSDR_SRC_CHANGENUMBER column, of numeric type. This column will contain system change number (SCN or LSN) from the source database which corresponds to the last replicated change for the record. The change number is SCN/LSN number of the commit operation of the transaction which contains the change, not the number under which the change has been written to transaction log.</p>
<b>source timestamp</b>	<p>Enables DBPLUSDR_SRC_TIMESTAMP column, of date/time type. This column will contain timestamp from the source database which corresponds to the last replicated change for the record. The change timestamp is the time of the commit operation of the transaction which contains the change, not the time when the change has been written to transaction log.</p>
<b>source transaction id</b>	<p>Enables DBPLUSDR_SRC_TRANSACTIONID column, of text type. This column will contain transaction id from the source database which corresponds to the last replicated change for the record. When the source database is Oracle, this means XID. When the source database is SQL Server, this column will contain 40-bit transaction identifier used in transaction log. This number <i>has no relation</i> with transaction number returned by CURRENT_TRANSACTION_ID function, nor with session number held by @@SPID system variable.</p>
<b>drop metadata columns that are no longer enabled</b>	<p>Let's assume metadata columns have been enabled in the configuration, then created in tables in the target database, and now some of those columns are being disabled. We could decide to:</p> <ul style="list-style-type: none"> <li>Remove the now disabled columns from tables in the target database, in this case we turn this option on,</li> <li>Keep the columns in tables in the target database, in this case we turn this option off.</li> </ul> <p>In the latter case the disabled columns will be preserved, but will no longer be updated.</p>

## Advanced configuration

### Replicator database objects

☐ Use default settings ☒ **Configure**



DBPlus Data Replicator creates a schema in the target database to keep utility functions and data which helps to keep track of the committed transactions. By default this schema is named DBPLUSDR\_{ReplicationName}.

Schema

Replicator database objects	
<b>Schema</b>	Name of the schema (user) which will be created in the target database to keep objects used by the replication mechanism.

### Reader

☐ Use default settings ☒ **Configure**



In parallel mode multiple transaction logs are processed simultaneously to improve replication throughput. The default auto setting will cause log reading process to switch to parallel mode when reading progress lags behind database activity.

**Run reader in parallel mode**

☒ auto ☐ always ☐ never

Max number of parallel workers

Min lag to switch to parallel  d  h  m  s



Invalid UTF-8 encoding detection should be enabled if you suspect that your database may contain data in character columns with UTF-8 encoding which does not form valid UTF-8 representation. This is possible when application uses UTL\_RAW functions in update/insert statements.


☐ Invalid UTF-8 encoding detection

Reader	
<b>Run reader in parallel mode</b>	<p>Parallel mode means that multiple threads are reading transaction log data from the source database at the same time in order to increase throughput. Reader in parallel mode can read simultaneously from multiple archive log files and, in addition to that, from current transaction log records (redo log). Reading threads are dynamically created and stopped depending on circumstances.</p> <p>In single mode reading is performed by a single thread working on continuous basis.</p> <p>The reader component switches between parallel and single mode depending on how much the reading process lags behind database activity, and on availability of archived logs.</p> <p><b>auto</b> The reader component always runs in parallel mode.</p> <p><b>always</b> The reader component always runs in parallel mode.</p> <p><b>never</b> The reader component always runs in single mode.</p>

<b>Max number of parallel workers</b>	Specifies maximum number of threads which read transaction logs from the source database which can be run simultaneously.
<b>Min lag to switch to parallel</b>	Specifies minimum amount of time by which the reading process has to lag behind current database activity before the reader component will consider switching to parallel mode (only if <b>auto</b> has been selected).
<b>Invalid UTF-8 encoding detection</b>	Turns on mechanism which detects and corrects problem of character columns with UTF-8 encoding containing binary data which do not constitute valid UTF-8 representation of character strings. This may happen if application writes binary data directly to a character column e.g. using UTL_RAW.CAST_TO_VARCHAR2 function. Detecting UTF-8 problems negatively affects performance of the transaction logs reading process.

**Applier**

☐ Use default settings   ☒ **Configure**

 Including all available columns in where clause of SQL commands will provide additional verification of correctness of the replication process. Limiting where clause to primary key columns will result in better performance.  
**In where clause of update/delete SQL commands include**  
☐ All available columns   ☒ Primary key columns only

☒ Enable bulk insert   Minimum number of operations merged

☒ Enable statement reordering

Preferred duration (millis) of a single execution bundle

**Parallel execution**  
☒ Disable  
☐ Enable schema level parallelism  
☐ Enable table level parallelism

Max number of parallel threads

Min lag to switch to parallel  d  h  m  s

<b>Applier</b>	
<b>Enable array bind</b>	<p>Turns on using of <b>array bind</b> feature available on Oracle database systems. Enabling this option leads to faster execution of sequences of insert/update/delete statements which share the same structure (differ only by values).  <b>Enabled</b> by default.</p> <p>Value in the <b>minimum number of operations merged</b> field specifies the minimum amount of atomic operations which will be merged into a single array bind invocation. The default value has been optimized for installations where the network latency between the replicator service and the target database is insignificant. The more the network latency the higher value of this parameter will be needed to achieve optimum performance.</p>
<b>Enable bulk insert</b>	<p>Turns on using of <b>bulk insert</b> feature available on SQL Server and PostgreSQL database systems. Enabling this option leads to faster execution of sequences of insert statements which share the same structure (differ only by values).  <b>Enabled</b> by default.</p>

	<p>Value in the <b>minimum number of operations merged</b> field specifies the minimum amount of atomic operations which will be merged into a single bulk insert invocation. The default value has been optimized for installations where the network latency between the replicator service and the target database is insignificant. The more the network latency the higher value of this parameter will be needed to achieve optimum performance.</p>
<b>Enable statement reordering</b>	<p>Turns on mechanism that reorders operations which are being applied on the target database in order to maximize the number of operations which can be executed in one go using either <b>array bind</b> or <b>bulk insert</b> feature. The reordering algorithm guarantees that standard integrity constraints will not be violated:</p> <ul style="list-style-type: none"> <li>• Primary and unique keys,</li> <li>• Foreign keys.</li> </ul> <p>If the target database uses other ways to enforce data integrity (e.g. implemented using triggers) then reordering operations may lead to runtime failures during execution of a reordered sequence of operations, in which case the original sequence of operations will be re-executed with optimizations <i>turned off</i>. If such situations happens often enough performance will drop compared to the performance achieved without statement reordering enabled.</p> <p>Option is available only if <b>Enable array bind</b> or <b>Enable bulk insert</b> is enabled. <b>Enabled</b> by default.</p>
<b>Preferred duration (millis) of a single execution bundle</b>	<p>Replicator does not execute a separate transaction on the target database for each transaction executed in the source database. As a way to speed up execution, updates made in different transactions are grouped together into “bundles”, each bundle is then executed within the scope of a single transaction in the target database.</p> <p>The parameter specifies preferred execution time (in milliseconds) of a single bundle. Larger bundle size means better potential for optimizations which leads to better performance; on the other hand bundle size which is too big may cause too many system resources to be used up on the target database and may also (on SQL Server and PostgreSQL database systems) lead to long lasting locks which negatively impact queries’ performance.</p> <p>Bundle size specified by this parameter is to be understood as a <i>maximum</i> bundle size – the parameter is effective only when number of operations awaiting for execution is big enough. If replication is performing just in time (with no significant lag) actual bundle sizes will depend on the number of operations awaiting for execution at any given moment.</p>
<b>Parallel execution</b>	<p>Enables parallel execution of operations in the target database in multiple threads. By default parallel execution is <b>disabled</b>. There are two modes of parallel execution available:</p> <p><b>Schema level</b> Operations can be executed in different threads if they affect different schemas.</p> <p><b>Table level</b> Operations can be executed in different threads if they affect different tables not connected by referential constraints (foreign keys).</p> <p>When executing in parallel mode a set of operations to be applied is divided into parts, each part is then executed in a separate thread, using a</p>

	<p>separate connection and a separate database transaction. Finalizing execution of a set of operations is performed in two phases:</p> <ol style="list-style-type: none"> <li>1. Await until all threads are done with executing their part,</li> <li>2. When all of the parts are ready to commit, each thread commits its transaction.</li> </ol> <p>Finalizing each commit does not occur at exactly the same moment, which means that in the target database, for a very short time, a state may occur which has never existed in the source database. If schemas are not interrelated schema level parallelization is safe. If table level parallelization is to be used you should consider possible consequences.</p>
<b>Max number of parallel threads</b>	Maximum number of threads executing updates on the target database in parallel mode. This value should match hardware resources available on the target database.
<b>Min lag to switch to parallel</b>	Allows to specify replication latency at which system switches to parallel execution mode. This parameter's purpose is to reconcile requirement of keeping up with the source database with avoiding possible side effects of parallel execution.

### System Resources

☐ Use default settings    ☒ **Configure**

**Transaction data memory pool**  
☐ auto    ☒ **set manually**

6 GB

System Resources	
<b>Transaction data memory pool</b>	<p>Specifies maximum allocation of physical memory (RAM) used by replicator service to store transaction data read from the source database. If switch is set to “auto” this value will be calculated as half of physically installed memory on the machine where the service runs.</p> <p>Replicator service will allocate as much physical memory for the purpose of storing transaction data as necessary, but no more than is specified here. Regardless, the service will allocate physical memory for other purposes.</p>

### Failed operations to ignore

**Ignore failed DROP command when dropped object does not exist for:**

☒ constraints  
☒ indexes

☒ Ignore failed ALTER command, when failure is due to the command having no effect, because the altered properties are already in the desired state.

This section allows us to select specific cases when the replicator will ignore replicated operations that failed to execute on the target database. This section will appear only if replication of DDL commands is supported.

<b>Ignore failed DROP command when dropped object does not exist</b>	Ignore DDL commands which attempt to remove a database object (DROP commands) that does not exist in the target database.
<b>Ignore failed ALTER command, when failure is due to the command having no effect, because the altered properties are already in the desired state</b>	Ignore DDL commands which attempt to change properties of a database object (ALTER commands), when the modified properties already have the desired values in the target database.

### Affected rows checking

By default when changes are applied to the target database replicator will check whether the right number of rows have been inserted/updated/deleted. However, when a particular table has NO UNIQUE INDEX this can lead to false positives. To avoid this from happening add problematic tables to the list of exceptions.

☐ Automatically add an exception when problem is detected

#### Affected rows checking exceptions

DELETE X

ADD +

Search

Schema	Name
SALES	CUSTOMERLOG

In this section we can enter exceptions to the use of checking number of affected rows as a way to verify correctness of the replication process. Normally each atomic update or delete operation will update or delete exactly one row of a given table. If, however, the table **does not have a unique index** then a single operation can update or delete multiple rows. Nevertheless the end result may (but will not necessarily) be correct. If you accept such a situation and do not want to augment the data model by adding a unique index to the table you should add the table to the list of exceptions. If the option **Automatically add an exception when problem is detected** is checked system will *automatically* add tables without unique indexes to the list of exceptions whenever the problem of a single operation affecting multiple rows is encountered.

### Logging

Diagnostic logs may help diagnose problems.

☐ Enable diagnostic logs

Logging	
<b>Enable diagnostic logs</b>	Turns on diagnostic logging. If enabled replication process will store key information about what is happening during replication processing in diagnostic log files. Sometimes this can help to diagnose problems. Size of the generated log files may reach gigabytes therefore it is not advised to enable this option without a good reason.

## Data conversion and data type mapping rules

### Data conversion rules

Data conversion rules control replication's behavior in a situation when data to be replicated does not fit into the target data model and in order to be entered into the target database requires truncation or rounding (incurring loss of precision). There are two possible causes:

1. Source and target databases run on different database management systems, data type on the target database which is a natural equivalent of certain data type on the source database have different range or precision. This specifically applies to date/time data types.
2. There are some differences in the data model between the source and target databases.

When such a case occurs, there are two possible actions to be taken:

1. Report an error, this means raising [an alert of type „Applier error“](#)
2. Perform truncation or rounding of the datum so that the trimmed value can be written to the column.

If database management system is the same in both databases it is assumed by default that data will always fit and no truncation or rounding is ever allowed. To setup data conversion rules anyway change this setting to **Setup data truncation and rounding options**.



Source and target databases run on the same RDBMS product. If the two databases use the same data model no data truncation or rounding will be necessary.

- ☒ Assume **data will fit** in the target database
- ☐ Setup data truncation and rounding options

By default all conversion problems cause an error to be reported except for problems concerning insufficient precision of numeric and data/time column types.

### Data conversion rules

When **text** is **too long** to fit the target column:

- ☒ Report **error**
- ☐ Truncate text to fit column

When one or more characters of **text** cannot be represented using the target column's **character set** :

- ☒ Report **error**
- ☐ Replace the characters with "?"

When **numeric value** is outside the **range** which can be represented by the data type of the target column:

- ☒ Report **error**
- ☐ Truncate to the nearest value



When **numeric value** cannot be represented with full **precision** by the data type of the target column:

- ☐ Report error
- ☒ **Round** to the nearest value

When **date/time value** is outside the **range** which can be represented by the data type of the target column:

- ☒ Report **error**
- ☐ Truncate to the nearest value

When **date/time value** cannot be represented with full **precision** by the data type of the target column:

- ☐ Report error
- ☒ **Round** to the nearest value

When **binary data** is **too long** to fit the target column:

- ☒ Report **error**
- ☐ Truncate data to fit column

When replicating from SQL Server to Oracle there arises a question of how to deal with empty character strings (character strings of zero length). SQL Server distinguishes between an empty string and NULL, Oracle on the other hand treats an empty string as synonymous with NULL. By default empty character strings from SQL Server will be replaced with NULLs. Instead you may opt to replace them with a special substitute text or to stop with an error whenever an empty character string is encountered in the replicated data.

Oracle database does not support **empty character string** (of zero length) and treats it as synonymous with NULL. When an empty string appears in the source data:

- ☐ Report error
- ☒ Treat it as **NULL**
- ☐ Replace it with the specified text

If the source database is an Oracle based database, which has been populated by replicating from an SQL Server based database, and if when replicating from the SQL Server empty character strings were replaced with a special substitute text in order to preserve differentiation between empty strings and NULLs, we will want to convert this special substitute text back into an empty string. To enable this a special data conversion option has been provided when replicating from SQL Server to Oracle. For example, if a tilde (~) has been chosen as an empty character string substitute configuration will look like this:

Oracle database does not support **empty character string** (of zero length) and treats it as synonymous with NULL. If a special text is used in the source database to represent an empty string it can be converted back into an empty string.

☒ Convert ~ into an empty string.

### Type mapping rules

Data type mapping is used in the following two situations:

1. When converting a DDL command to execute on the target database. This applies to replications from SQL Server to Oracle, but also when replicating from SQL Server to SQL Server and from



Oracle to Oracle in CUSTOM mode it is possible to define customized type mapping rules, which will be used to convert data types.

2. When constructing table definition to be created in the target database in the process of automatic initial data transfer. This applies to all source and target database combinations.

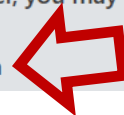
When replication is using CUSTOM mode (available for replication between Oracle databases) type mapping is *disabled*.

If source and target databases are based on the same database management product original data types will be preserved by default, type conversion can be enabled by selecting **enable type conversion** option.



Source and target databases run on the same RDBMS product. However, you may still configure conversion between data types.

☐ preserve original types ☒ **enable type conversion**



Type conversion table contains (specific for each combination of database management products) predefined source data types and suggested target data types. Target data types can be modified.


	NUMBER (P, S)	decimal (P, S)
	NUMBER	<div>decimal (P, S)</div> <div>P = 38</div> <div>S = 10</div>
	BINARY_FLOAT	real

When editing target data type select the desired data type from the list and then, if the selected type is a parametrized data type, enter formulas for calculating its parameters' values. A formula may be a constant, it may refer to parameter of the source data type, or it may be an arithmetic expression. For example, we may map text type varchar(N) into a text type with twice the maximum length of the original by entering  $2 * N$  as the formula for parameter N, which will produce the following entry in the mapping table:

	varchar (N)	varchar (2 * N)
--	-------------	-----------------

## Configuration of optional alerts


### Archive log availability alert

 Alert when the amount of archive logs between the position currently being read and the start of available archive logs is less then the specified time span. When this amount goes down to zero required archive logs become unavailable and the replication stops.

☒ enable

Alert when less then 00 d 03 h 00 m 00 s is left

### Replication lag alert

 Alert when replication lag is greater than the specified time span. The lag being measured is "update to write", meaning the difference between the time the last replicated change appeared on the replica and the time it was performed on the source.

☒ enable

Alert when lag is greater than 00 h 01 m 00 s

Most alerts are generated always when specific circumstances arise and cannot be configured by the user. The following alerts can be enabled and configured:

<b>Archive log availability alert</b>	This alert is generated when position currently being read is close to the start of available archive log (Oracle) or the first change available in the change archive (SQL Server). When reading process passes beyond the start of available logs the replication stops. If the source database is SQL Server then enabling this alert only makes sense if disk space limit or data retention limit has been enabled in Change Capture component setup.
<b>Replication lag alert</b>	This alert is generated when replication latency is greater than the specified time span. Replication latency is calculated as difference between the time of execution in the target database and the time of execution in the source database of the last applied change.

## Planned replicator breaks schedule

Planned breaks schedule allows to define time periods during which replicator will be inactive.

There are four repetition modes available:

<b>one time</b>	One time break, in this mode we specify: <ul style="list-style-type: none"> <li>Start date and time of day,</li> <li>End day and time of day.</li> </ul>
<b>daily</b>	Break will be activated each day, in this mode we specify: <ul style="list-style-type: none"> <li>Start time of day,</li> <li>End time of day.</li> </ul>
<b>weekly</b>	Break will be activated each week, possibly multiple times, in this mode we specify: <ul style="list-style-type: none"> <li>Start days of the week,</li> <li>Start time of day,</li> <li>End day and time of day.</li> </ul>
<b>Monthly</b>	Break will be activated once each month, in this mode we specify: <ul style="list-style-type: none"> <li>Start day of month,</li> <li>Start time of day,</li> <li>End day and time of day.</li> </ul>

For months which have less days than the day specified as start day of month the *last day* of month will be used.

Break start time can be configured in one of two ways:

<b>real time clock</b>	Break starts on the date and at the time specified in the schedule, respective to the selected time zone.
<b>replica timestamp</b>	Break starts at the moment the replica reaches state corresponding to the state of the source database at the time point specified in the schedule.

Break end time can also be configured in one of two ways:

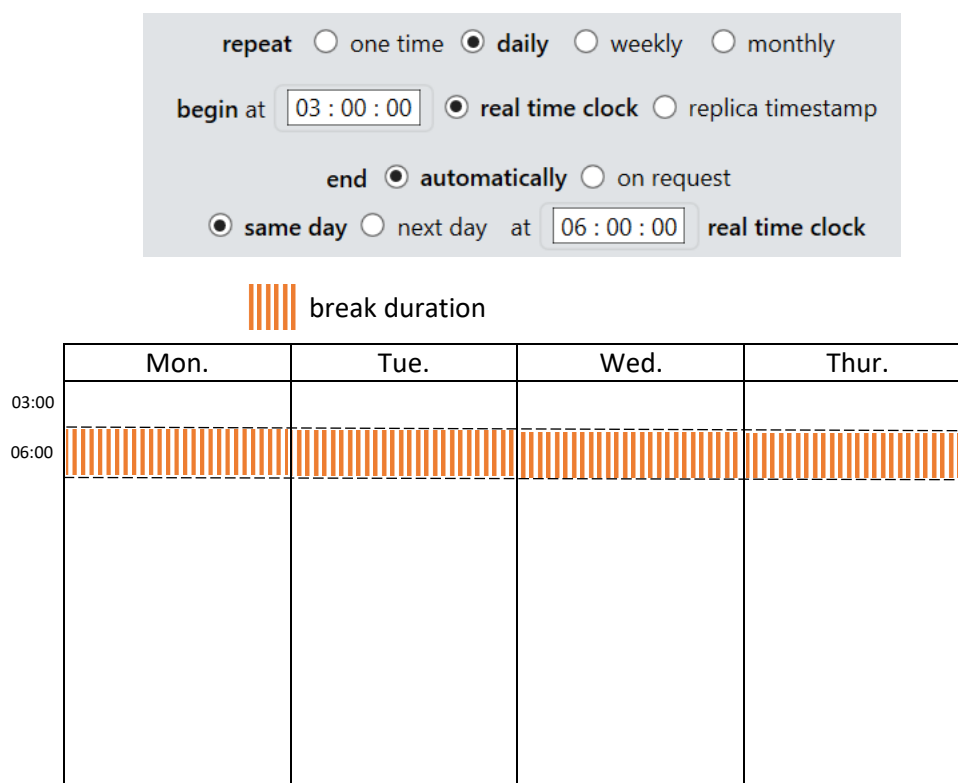
<b>real time clock</b>	Break ends on the date and at the time specified in the schedule, respective to the selected time zone.
<b>on request</b>	There are two methods to end the break: 1. In the GUI client using the TERMINATE button, 2. By means of a script or command issued on the target database, using API provided for this purpose in the target database.

This gives total of four different ways to configure a break schedule. Each one has its own characteristics and corresponds to a different use case.

*Start by real time clock, end by time clock*

Schedule defines a series of “windows” when replication takes a break. During time spans specified by the windows replication does not work, outside of them replication works.

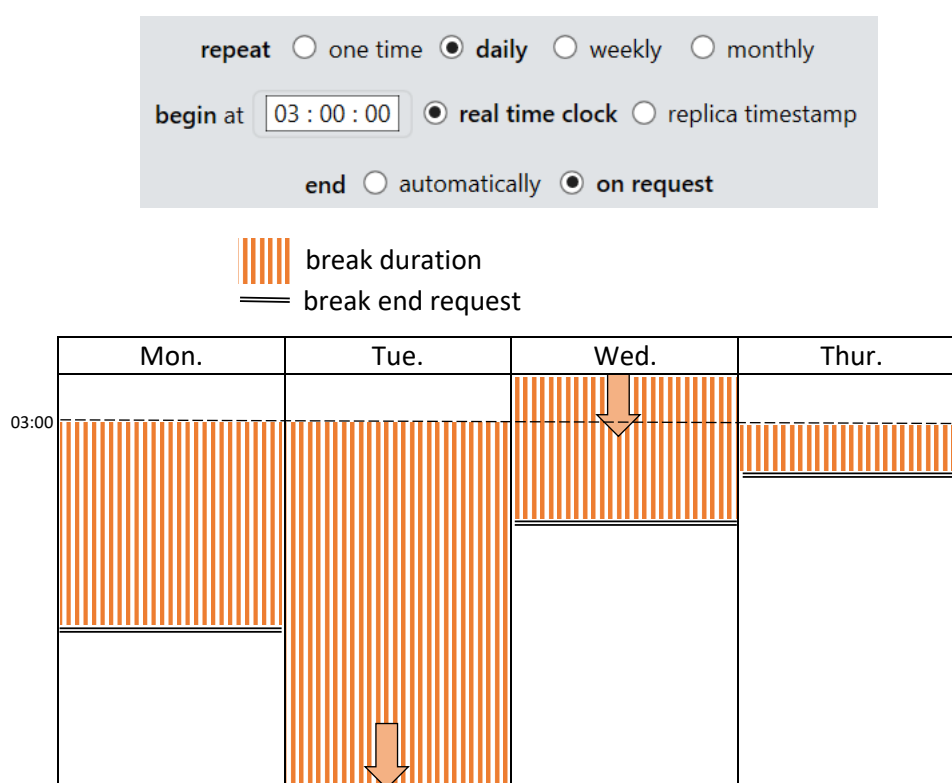
*Example*



### Start by real time clock, end on request

Break starts on the date and at the time specified in the schedule, ends when end is requested. If the request comes with a long delay, after the time when the *next* planned break is due, this next break will not be activated.

### Example



Because request to end the break which started on Tuesday at 3:00 arrived after 3:00 on Wednesday, the break due in Wednesday was been activated.

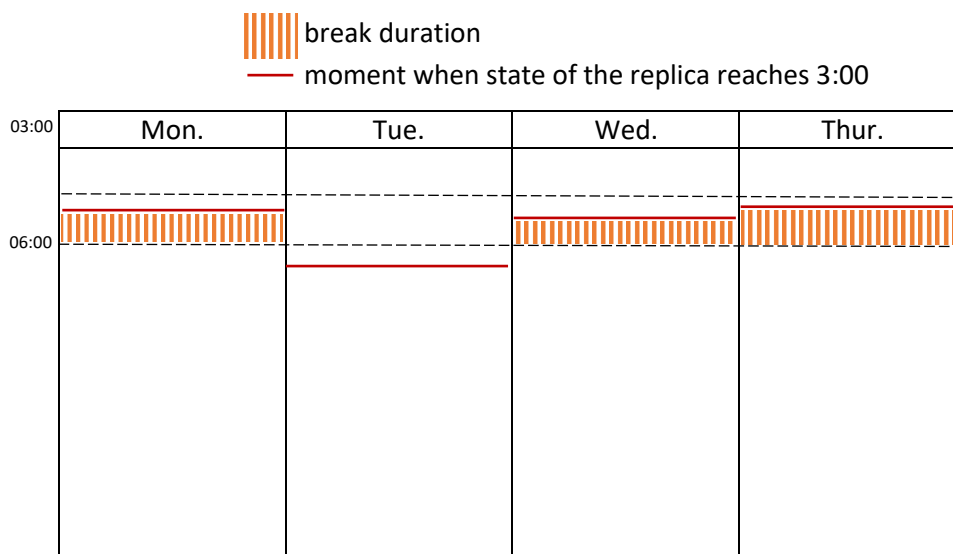
### Start by replica timestamp, end using real time clock

Break starts at the moment the replica reaches state corresponding to the state of the source database at the time point specified in the schedule as break's start, ends on the day and at the time configured in the schedule as the break's end. If replication has big enough lag, so that the replica reaches state corresponding to the time point configured as break's start only *after* the time configured as break's end, the break will not be activated. (This situation may be interpreted as the break being activated and then immediately terminated, as the ending condition has been met.)

### Example

The configuration interface shows the following settings:

- repeat:** ☐ one time ☒ daily ☐ weekly ☐ monthly
- begin at:**  ☐ real time clock ☒ replica timestamp
- end:** ☒ automatically ☐ on request
- same day** ☒ **next day** ☐ at  **real time clock**



On Tuesday the replica reached the 3:00 state after 6:00, therefore no break was activated on Tuesday.

*Start by replica timestamp, end on request*




Break starts at the moment the replica reaches state corresponding to the state of the source database at the time point specified in the schedule as break's start, ends when requested. In this case there is no (direct) connection between duration of breaks and the real time. When state of the replica reaches a time point designated as break's start in the schedule, break starts and continues until its end is requested. Then replication continues until the *next* time point designated as break's start is reached, whereupon next break is activated, etc.

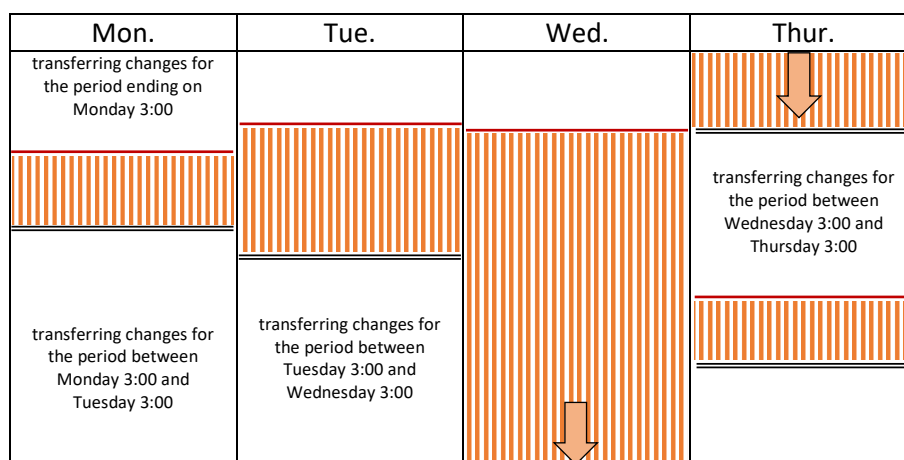
*Example*

repeat ☐ one time ☒ daily ☐ weekly ☐ monthly

begin at  ☐ real time clock ☒ replica timestamp

end ☐ automatically ☒ on request

 break duration  
 moment when state of the replica reaches 3:00  
 break end request



### Time zone

Because the time zone of the database may be different from the time zone on then replication server, and this in turn may differ from the time zone on the client, the schedule settings contains time zone setting. Wherever we refer to real time, date and time is interpreted as being defined in the selected time zone.

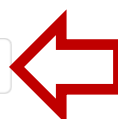
### Terminating a break

You can terminate a break once it has been activated using the **TERMINATE** button, which will appear on the DASHBOARD tab page.



Replication MyReplication is not active due to a planned break triggered by schedule "MySchedule". The break ends on 2019-10-18 18:00:00.

TERMINATE X



There are two reasons why this button may be used:

1. To end a break ending on request.
2. To end a break scheduled to end at a specified time, after having made changes to the schedule. Modifications made to a break schedule impacts *future* breaks, but do not impact a break that has already been activated. At the moment a break is activated the date and time on which it will be terminated is calculated, this date and time does not change.

## 6.7 Maintenance tools





Maintenance tools are available under **TOOLS** button the replication management panel.

## Tool for generating scripts to transfer data using data pump

Tool is available only for replications which replicate data between two Oracle databases. It produces batch scripts for use on Windows operating system. The scripts execute data export from the source database and data import to the target database using data pump tool which is available on Oracle database systems.

For each *schema* targeted for transfer a separate data transfer script is generated. If data from multiple schemas are to be transferred you can reduce transfer time by executing multiple scripts in parallel.

 **CREATE DATA PUMP SCRIPT**

 **VIEW PREPARATION REPORT**

### EXPORT

database

myservice:1522/SOURCE

user

C##REPLUSER

password

Enter password for source database

directory

Default is DATA\_PUMP\_DIR

☐ Init directory as

☒ Export current content
 ☐ Specify flashback SCN

### IMPORT

database

myservice:1522/TARGET

user

TESTDATA

password

Enter password for target database


directory

Default is DATA\_PUMP\_DIR

☐ Init directory as

### OPTIONS

☐ Include constraints




When creating tables in target database alter all character data columns with length specified in bytes to length specified in characters, eg. VARCHAR2(10 BYTE) becomes VARCHAR2(10 CHAR).

☐ Upgrade character data column length from BYTE to CHAR

### SCRIPTS

Save scripts in

C:\Temp\dump

 **CREATE SCRIPTS**

EXPORT	
<b>Database</b>	Connect identifier of the source database.
<b>User</b>	User name in the source database. [1]
<b>Password</b>	User's password in the source database. [1]
<b>Directory</b>	Name of a directory object in the source database which serves as a reference to the location in the file system where the exported data will be stored. [2]
<b>Init directory as</b>	Can be used to enter path to the folder where the exported data will be stored. The path must be accessible from the machine where the source database works. [2]
<b>Export current content</b>	The current state of the data at the moment scripts are run will be exported [3].

<b>Specify flashback SCN</b>	Use this option if you want the exported data to represent the state of the source database at a specific moment in time.
<b>IMPORT</b>	
<b>Database</b>	Connect identifier of the target database.
<b>User</b>	User name in the target database. [1]
<b>Password</b>	User's password in the target database. [1]
<b>Directory</b>	Name of a directory object in the source database which serves as a reference to the location in the file system from which the imported data will be loaded. [2]
<b>Init directory as</b>	Can be used to enter path to the folder from which the imported data will be loaded. The path must be accessible from the machine where the target database works. [2]
<b>OPTIONS</b>	
<b>Include constraints</b>	Determines whether constraints will be included.
<b>Upgrade character data column length from BYTE to CHAR</b>	If this option is turned on all columns with character data types which in the source database have length specified in bytes (BYTE) will be converted to columns with length specified in characters (CHAR) in the target database. E.g. VARCHAR(50 BYTE) will become VARCHAR(50 CHAR) [4]
<b>SCRIPTS</b>	
<b>Save scripts in</b>	Folder where scripts generated by this tool will be saved.

[1] If value is not entered here you will be asked to enter it during script *execution* time. Because for each schema a separate script is generated we will have to enter the value multiple times if data from multiple schemas is being transferred. Therefore it is more convenient to enter the values on the script generation form.

[2] The data pump tool gets name of a *directory* as a parameter. Directory is in turn associated with a path pointing to a folder in the file system. This means there are two possible ways to deal with the configuration of data storage path:

1. In the **Directory** field, enter name of the directory which has been associated (or you plan to associate later) with the desired path,
2. In the **Directory** field, enter directory name and in the **Init directory as** field enter the desired path, the directory will be associated with the path entered in the *Init directory as* field when the scripts are *executed*.

[3] When first script is executed for the first time the current SCN of the source database will be fetched and written to a file named FlashbackScn.txt. Value written to this file will be then used by all the subsequently executed scripts. If you want to "reset" the SCN number all you have to do is to remove the file.

[4] This conversion prevents later problems in a situation when character encoding is different between source and target databases. As an example let's assume that the source database uses EE8MSWIN1250 encoding and the target database uses AL32UTF8 encoding. In a column of type VARCHAR(5 BYTE) in the source database value „zóty” has been written. The value fits in the column because it requires 5 bytes of storage in EE8MSWIN1250 encoding. On the other hand in the target database with AL32UTF8 encoding value of „zóty” requires 8 bytes of storage (because the letters „ż”, „ó” and „t” of the Polish alphabet use up 2 bytes each when encoded as UTF-8) and will not fit in a column of type VARCHAR(5 BYTE). The solution is to change the column type to VARCHAR(5 CHAR)



which will cause Oracle to allocate adequate amount of storage to hold *any* 5 characters in AL32UTF8 encoding.

Data import using scripts generated by this tool works together with the option **automatically set by import** in start point settings. If the *automatically set by import* option has been selected the replication process will set start point to the flashpoint SCN used to export the data which has been most recently imported. If replication is already running and you want to update the target database using the data import tool and then restart the replication with the new start point do the following:



1. Stop the replication service.
2. Use the button



on the [setup screen](#) and apply the change (option of *automatically set by import* remains selected). Replication state will be reset and the replication will start from clean slate with a new start point when replication process is started again.

3. Perform data import to the target database.

After replication process has been started the new start point will be set to flashpoint SCN associated with the most recent data import.

When the tool is run a number of files is generated and stored in the folder specified in the **save scripts in** field. The following files are created:

1. For each schema 3 scripts:
  - a. export-SCHEMAT.bat (performs export)
  - b. import-SCHEMAT.bat (performs import)
  - c. transfer-SCHEMAT.bat (performs export and then import)
2. If data from more than one schema is transferred an additional script is created with name TransferAll.bat
3. File ReadMe.txt
4. Subfolder utils holds miscellaneous helper bat and sql scripts (which are not intended to be executed on their own)

Log files created when scripts are executed are written to a subfolder named log. For each of the export-\*.bat and import-\*.bat scripts a separate log file is created.

## Tool for viewing the replication preparation report

CREATE DATA PUMP SCRIPT

VIEW PREPARATION REPORT

ORIGINAL CURRENT REFRESH

### SUMMARY

Number of tables 1  
Data volume 0 B  
Index volume 0 B

### SYSTEM CHECK RESULTS

Some of the tables selected for replication have no unique index. This may lead to runtime problems.  
TESTDATA.MAMOKO

Archive logs are available  
Archive logs are available for the period of 15 days.

Report created on 2019-06-11 16:14:45

Here you can view the replication preparation report that has been created during the process of creating the replication. The same kind of report but referring to the *current* configuration and *current* state of the source and target databases can be obtained by selecting option **CURRENT**.

## Change Capture Component maintenance tool

VIEW PREPARATION REPORT

CHANGE CAPTURE COMPONENT MAINTENANCE

CHANGE CAPTURE JOB STATUS

Component is installed and active

Component is installed in database: dbpluscc

### CHANGE ARCHIVE OPTIONS

Records of captured changes will be stored on disk in archive files. Folder must be accessible from the machine where the database server runs.  
archive folder C:\DCD\StageFiles

If this option is selected, data in the change archive will be automatically deleted once it has been "consumed" by replication process. This will save space but will also prevent from restarting replication from an earlier point.  
☐ Automatically truncate change archive

This option specifies for how long data will be preserved in the change archive.  
☒ Limit data retention to 01 d 00 h 00 m

If this option is selected, data in the change archive will be deleted as necessary to keep disk usage below the specified maximum. This setting is per database, if more then one database is set up as a replication source each may use up to the amount of disk space specified here.  
☒ Limit disk space usage to 1024 MB

### SYS ADMIN CREDENTIALS

System administrator credentials are required in order to install or uninstall the component.

authentication Database

user

password

CONFIGURE STOP UNINSTALL

DBPLUS Change Capture Component is based on CDC (Change Data Capture) which is a standard feature of SQL Server. Installing the component is necessary for replications using SQL Server databases as source to function.

The component consists of a set of libraries (dll files) running on .NET 4.0 platform, T/SQL functions and procedures, and configuration tables. All these object are kept in a single databases created solely for that purpose. From administrative point of view the component is simply a database.

Basic options (available for installation):

<b>database name</b>	Name of the database where the component will be installed.
<b>data file</b>	Name of data file for the database where the component will be installed.
<b>log file</b>	Name of transaction log file for the database where the component will be installed.

Change archive options:

<b>archive folder</b>	Change archive folder location. In the process of capturing changes using the component change information is written to disk. Repository of change information stored in files is called „change archive“. The parameter specifies the archive's location. Remember this location <i>must be accessible for the SQL Server instance</i> .
<b>Automatically truncate change archive</b>	If this option i set data in the change archive will be delete right after they have been processed all relevant replication processes. This leads to lower disk usage but makes it impossible to restart replication with start point LSN set to less the last LSN processed.
<b>Limit disk space usage to</b>	Allows to specify maximum disk space used by the change archive. Note: If replication process is lagging behind and description of changes which have been executed in the database between the point being currently processed by replication process and now happen to take up more disk space then specified here, it is possible for changes to be deleted from disk before replication process has a chance to process them, which, in effect, will cause replication to fail.
<b>Limit data retention to</b>	Allows to specify maximum time span for which data will be retained by the change archive. Note: If replication process is lagging and the lag is greater than time span specified here, it is possible for changes to be deleted from disk before replication process has a chance to process them, which, in effect, will cause replication to fail.

Available operations:

<b>CONFIGURE</b>	Applies changes in the parameters described above.
<b>STOP</b>	Stop the component. If component is stopped change capture stops in all databases where change capture is enabled (from which data is replicated). It is also not possible to enable, modify or disable change capture, and reading from change archive. Stopping the component causes all replications which have as a source a database within given SQL Server instance to stop reading captured changes.
<b>RESUME</b>	Resume component which has previously been stopped.
<b>INSTALL</b>	Installs the component.
<b>UNINSTALL</b>	Uninstalls the component.

	Uninstalling the component causes all replications which have as a source a database within given SQL Server instance to stop reading captured changes. After the component is reinstalled replications will resume.
<b>UPGRADE</b>	Upgrades the component to the current version (compatible with the version of DBPLUS Data Replicator).

All operations on Change Capture component require credentials of a user with system administrator rights (a user with sysadmin role).

If the source database belongs to an Always On Availability Group component should be upgraded and configured on all servers belonging to the group, otherwise replication will stop after failover switch. By default each operation is executed on all servers belonging to the Always On group. If necessary specific servers may be excluded using the **include** check box.

The replicated database belongs to **Always On Availability Group** AlwaysOnDbplus. DBPLUS Change Capture Component should be installed on all nodes of the group to ensure correct functioning of replication process after switch occurs.

- Server **DBPLUSSQLSRV1\SQLNODE1** ☒ include
- ☒ Component is installed and active
- Server **DBPLUSSQLSRV2\SQLNODE2** ☒ include
- ☒ Component is installed and active



Displayed parameters come from the active replica. Status „stopped” is displayed when component on the active replica is “stopped”.

If replication manager is unable to connect to database servers using names displayed here (DBLPUSSQLSRV1\SQLNODE1 or DBLPUSSQLSRV2\SQLNODE2 in the above example) please apply solution described [here](#).

## Monitoring change capture job status

In databases running on SQL Server capturing changes for the purposes of replication is performed by a job (which is executed in the background by SQL Server Agent). One change capture job is generated per each database used as a source database by a replication in DBPLUS Data Replicator. Change capture job periodically executes a process which extracts changes from the transaction log (using CDC to accomplish this) and writes the results to disk.

VIEW PREPARATION REPORT

CHANGE CAPTURE COMPONENT MAINTENANCE

CHANGE CAPTURE JOB STATUS

job name

status

last processed LSN

last processed timestamp

last execution time

last execution status

dbpluscc.BORSUK\_capture

RUNNING

0x00000917:00003B7F:0001

2019-10-28 09:29:56

2019-10-28 09:29:56

OK

START

STOP

<b>job name</b>	Name of the job.
<b>status</b>	One of: <ul style="list-style-type: none"> <li>NOT INSTALLED</li> </ul>

	<ul style="list-style-type: none"> <li>• STOPPED</li> <li>• RUNNING</li> </ul>
<b>last processed LSN</b>	LSN of the last processed change from the transaction log.
<b>last processed timestamp</b>	Timestamp of the last processed change from the transaction log.
<b>last execution time</b>	The last time transaction log analysis process was run.
<b>last execution status</b>	Whether the last transaction log analysis process executed without errors.

## 6.8 Restarting replication from a new start point

Restarting replication from a new start point means starting replication from a from a clean slate, while retaining all settings and history. To do this click CONFIG button on the replication management panel and then select **START & END** tab.

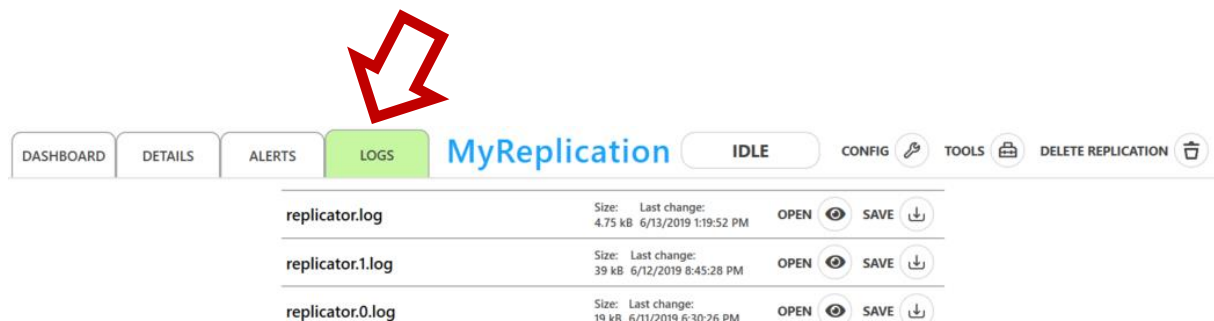
Start point options can be modified as long as the replication has not yet been started for the first time. After replication has been started a relevant information and the **RESTART REPLICATION** button will appear.



After clicking the **RESTART REPLICATION** button start point configuration options will appear. Rules for setting up start point here are identical to those during creation of a new replication as [described here](#).

When using automatic initial load feature there is no need to use the replication restart feature. The preferred way to reinitialize tables in the target database with fresh data from the source database is to use the **REINITIALIZE** command on all tables showing on [the replicated tables list](#).

## 6.9 Viewing the logs

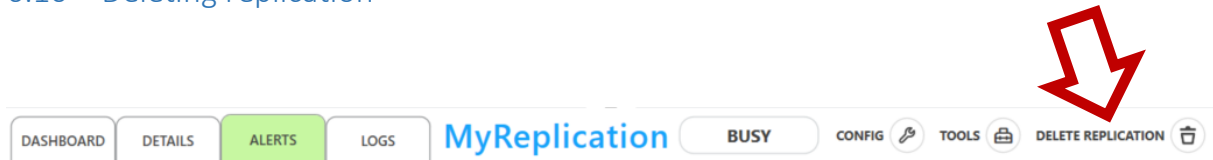


In order to access replicator's logs select the **LOGS** tab.

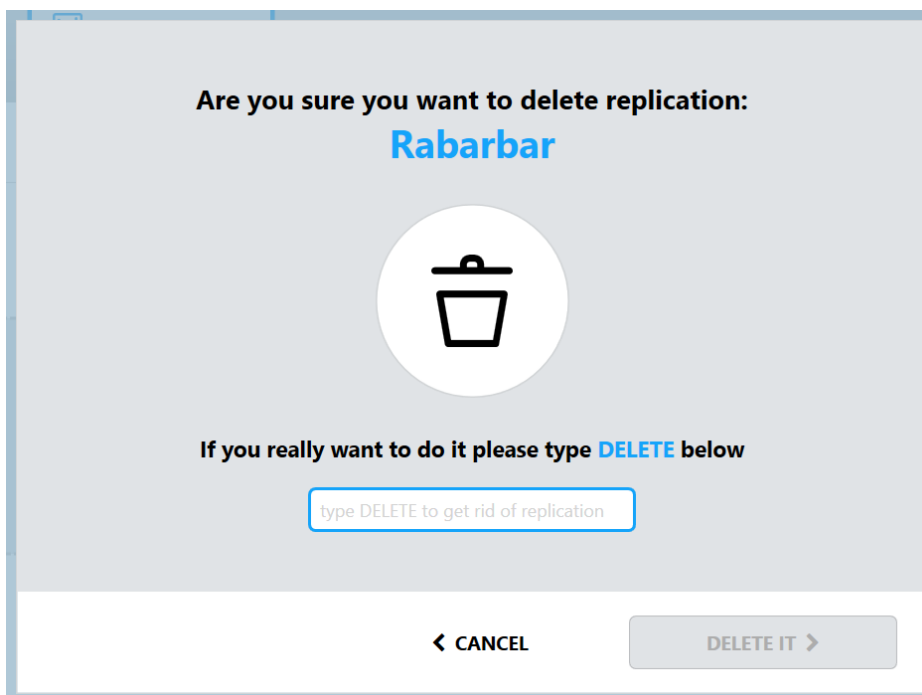
Files named replicator\*.log contain information about starting, stopping and other changes of the replication's state, errors, warnings and other important information.

Files named diagnostic\*.log are created only if option **Enable diagnostic logs** has been enabled in advanced replication settings. They contain dump of key information about what is happening during the replication process.

## 6.10 Deleting replication



In order to delete replication click on **DELETE REPLICATION** button on the replication management panel.




Replication will be deleted after deletion is confirmed by typing **delete** into the text field and clicking on **DELETE IT** button.

If the replication being deleted had SQL Server database set as source and if there are no more replications using this database as a source the following box will appear:

Replication **RABARBAR** has been removed. There are some related issues which require your attention.


If you no longer need CDC in database SourceDB you should disable it. Sys admin credentials are required. The following commands will disable CDC for database SourceDB:  
 use [SourceDB];  
 EXEC sys.sp\_cdc\_disable\_db;  
 ... or disable CDC now using the tool below:

 **sys admin credentials**

authentication

user name

password

 **DISABLE CDC**

**OK**

Disabling CDC in a database requires system administrator's privileges (sysadmin role), therefore it is not performed automatically during replication deletion. It can be done using an appropriate command or by clicking **DISABLE CDC** button, after entering credentials of a user with system administrator privileges.

## 7 Additional information

### 7.1 Information about replication status accessible in the source database

In the target database a table is maintained which holds information about current replication status. The table is named DBPLUSDR\_{replication name}.ReplicationStatus (schema name may be different if it has been changed in the advanced replication options). This table holds exactly one record (or zero, if it has not yet been initialized) and consists of the following columns:

<b>SourceSystemTime</b>	Current system time in the source database at the moment of writing the record.
<b>TargetSystemTime</b>	Current system time in the target database at the moment of writing the record.
<b>AppliedTimestamp</b>	Timestamp (according to the clock in the source database) of the change last applied in the target database.
<b>AppliedChangeNumber</b>	SCN (System Change Number) of the change last applied in the target database.
<b>ReplicaTimestamp</b>	The latest time point (according to the clock in the source database) which the data in the target database represents. If the replication is active then ReplicaTimestamp = AppliedTimestamp. If replicator does not have anything to do for a while (because there are no new changes to replicate) AppliedTimestamp remains unchanged while ReplicaTimestamp is periodically updated with the latest time point which is known not to contain any changes not replicated to the target database.
<b>ReplicaChangeNumber</b>	The latest change number (SCN or LSN) which the data in the target database represents. If the replication is active then ReplicaChangeNumber = AppliedChangeNumber. If replicator does not have anything to do for a while (because there are no new changes to replicate) AppliedChangeNumber remains unchanged while ReplicaChangeNumber is periodically updated with the latest change number which is known not to contain any changes not replicated to the target database.
<b>Status</b>	Status of the replicator process, see table below for more details.
<b>BreakEndTime</b>	Contains planned end time of the current break when Status = BREAK.

Data written to this table makes it possible to calculate current replication delay (how much current state in the target database lags behind the state of the source database). Current delay can be calculated from the formula:

$$\text{SourceSystemTime} - \text{ReplicaTimestamp} + \{\text{Current system time}\}^1 - \text{TargetSystemTime}$$

The part SourceSystemTime - ReplicaTimestamp is replication delay at the moment the record was written, the part {Current system time} - TargetSystemTime is the time which has passed since the record was written. Increase of the delay calculated using this method may be caused by the replication process not being able to cope with the amount of data to replicate or, on the other hand, by the fact that nothing has been happening in the source database for a considerable amount of time.

<sup>1</sup> LOCALTIMESTAMP in Oracle database



The point in time which the current state of the target database represents adjusted to the clock of the target database can be calculated as follows:

$\text{ReplicaTimestamp} + \text{TargetSystemTime} - \text{SourceSystemTime}$

Possible values in the Status column:

<b>RUNNING</b>	Replicator process is working. It should be noted that value RUNNING in the Status column does not guarantee that the replicator process is actually working, as it might happen that the value could not have been updated due to a malfunction. To verify that the replicator is indeed running the TargetSystemTime column should also be checked. The time that have passed since the last update should not exceed a few minutes.
<b>STOPPED</b>	Replicator process is stopped.
<b>BREAK</b>	A <a href="#">planned break</a> is in effect.
<b>FINISHED</b>	The replication end point has been reached.

## 7.2 Database level API for managing planned breaks

The planned breaks API will be available on each database configured as target database of a replication running on DBPLUS Data Replicator. It consists of a set of procedures and functions located in schema named DBPLUSDR\_{REPLICATION NAME} (schema name may be different if it has been changed in the advanced replication options). In Oracle these procedures and functions will be part of a package named **replication\_utils**, in SQL Server and PostgreSQL, as these database management systems do not support packages, they will be created directly within the schema.

### is\_break\_active

Checks whether the replication currently is in a “planned break” state.

In Oracle the return value is of type NUMBER and has a value of 1 (true) or 0 (false). In SQL Server it is of type BIT and has a value of 1 (true) or 0 (false). In PostgreSQL it is of type BOOLEAN.

### is\_break\_end\_requested

Checks whether the replication currently is in a “planned break” state and break end has been requested. Replication may remain in this state for a long time if the replication service is not running.

In Oracle the return value is of type NUMBER and has a value of 1 (true) or 0 (false). In SQL Server it is of type BIT and has a value of 1 (true) or 0 (false). In PostgreSQL it is of type BOOLEAN.

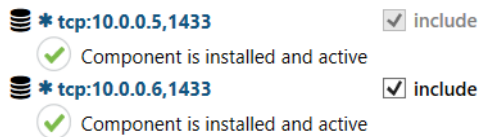
### request\_break\_end

Registers request to end planned break. Its work consists of setting a flag in a record kept in the database, which means its execution is instantaneous. After the flag has been set the replication service will, provided it is running and is able to access the database, read it and terminate the break.

If the replication is not in a “planned break” state, or the break is not configured as terminated on request, an *error will be raised*.

### 7.3 Manually overriding server addresses of Always On Availability Group nodes

Server names used to connect to Always On Availability Group nodes are retrieved automatically from the Always On configuration. In a typical scenario, provided that replication manager service is installed in the same domain as database servers, there should be no problem connecting to database servers using these names. If, however, this turns out to be impossible you can simply *edit them* by clicking them and replacing with an address to be used by replication manager service to connect to the given database server. When server address has been edited a star will appear on the left as shown on the below image:



After server address has been edited and an operation (e.g. component installation) executed the address will be stored by the replication manager as the address to be used when connecting to the node, so there will be no need to reenter it.

## 8 Restrictions and limitations

### 8.1 Products

<b>Source database</b>	Oracle release 10.2 and later, versions Standard and Enterprise.
<b>Target database</b>	Oracle SQL Server PostgreSQL

### 8.2 Oracle as source database

#### 8.2.1 Supported data types (source database)

CHAR NCHAR VARCHAR VARCHAR2 NVARCHAR2 LONG LONG RAW CLOB NCLOB BLOB NUMBER BINARY_FLOAT BINARY_DOUBLE DATE TIMESTAMP TIMESTAMP WITH TIME ZONE TIMESTAMP WITH LOCAL TIME ZONE INTERVAL YEAR TO MONTH INTERVAL DAY TO SECOND	YES
XMLTYPE	Oracle release 11.2.0.3 and later
ADT (abstract data types, also known as user data types) SDO_GEOMETRY, SDO_TOPO_GEOMETRY	Oracle release 12 and later
ROWID UROWID REF BFILE ORDDICOM ANYDATA ANYDATASET ANYTYPE MLSLABEL URITYPE SDO_GEORASTER VARRAY (collections)	NO

Tables using compression are not supported in Oracle release 10.

Replicating data in tables with NOLOGGING options enabled will not be possible if BULK operations are used.

### 8.2.2 Restrictions for CUSTOM mode

In CUSTOM mode user can specify objects and operations to replicate.

In this mode the following objects are supported:

1. Data
2. Table definitions
3. Indexes
4. View definitions
5. Packages, procedures and functions

### 8.2.3 Restrictions for EXACT mode

In EXACT mode all operations are replicated.

All **TRIGGER** objects are **disabled**.

### 8.2.4 Ignored schemas

Some of the predefined database schemas are unconditionally left out for replication purposes:

AUDSYS  
 TSMSYS  
 PERFSTAT  
 AWR\_STAGE  
 CSMIG  
 APPQOSSYS  
 OJVMSYS  
 DVSYS  
 TRACESVR  
 ORACLE\_OCM  
 DBSNMP  
 ANONYMOUS  
 CTXSYS  
 DBSNMP  
 EXFSYS  
 LBACSYS  
 MDSYS  
 MGMT\_VIEW  
 OLAPSYS  
 ORDDATA  
 OWBSYS  
 ORDPLUGINS  
 ORDSYS

OUTLN  
SI\_INFORMTN\_SCHEMA  
SYS  
SYSMAN  
SYSTEM  
WK\_TEST  
WKSYS  
WKPROXY  
WMSYS  
XDB

## 8.1 SQL Server as source database

The following operations are replicated:

1. Data manipulation: INSERT, UPDATE, DELETE
2. DDL operating on tables:
  - a. Adding tables (CREATE TABLE)
  - b. Removing tables (DROP TABLE)
  - c. Changing name (sp\_rename)
  - d. Changing primary key (ALTER TABLE ADD/DROP ... PRIMARY KEY)
3. DDL operating on table columns:
  - a. Adding columns (ALTER TABLE ADD)
  - b. Removing columns (ALTER TABLE DROP COLUMN)
  - c. Changing column type (ALTER TABLE ALTER COLUMN)
  - d. Changing column's *nullable* option (ALTER TABLE ALTER COLUMN)

The amount of data inserted into LOB columns in replicated tables is limited by global parameter **max text repl size**, the default value being 65536 bytes.

<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-max-text-repl-size-server-configuration-option?view=sql-server-2017>

If a table is replicated by CDC one cannot:

1. Change its columns' names,
2. Use TRUNCATE command against it.

CDC may not function properly if database engine service or „SQL Server Agent“ service runs on **network service** account, this may lead to error 22832.