DBSync

User Manual

V2.2

2025.5

1 What's DBSync

DBSync is a lightweight desktop software used for database comparison and synchronization, it can find out the differences between two databases and synchronize them to keep both identical.

2 Main Features

- **Independent and Non-Interfere**

  DBSync runs independently, you need neither to set script or trigger in databases nor to change existing software, you just need to connect databases and set the fields mapping, then sync can be done.

- **Support a Variety of Databases**

  DBSync supports a variety of databases including Oracle, MySQL, SQL Server, Access, PostgreSQL, MongoDB, DB2, ES, SQLite, Firebird, etc.

- **Support Heterotype Sync**

  Besides the sync between same type of DB, DBSync can also sync between different type of DB, such as SQL Server to MySQL, MySQL to Oracle, etc.

- **Remote and Cross-Platform Sync**

  Besides the sync between two local DB, DBSync can also sync between local DB and remote DB, as long as they can be connected the sync can be done.

- **Support Incremental Sync**

  Besides the sync of full data, DBSync can also sync only the incremental data, it can find out the differences between two DB then sync them. This makes the process more efficient.

- **Delimit Sync Range**

  You can designate the range involved in sync task, it can be either the whole DB or some tables in DB or some rows & fields in table, you can delimit which data to sync.

- **Support 1-to-N and N-to-1 Sync**

  Some company has a headquarter DB and several branch DB, DBSync can distribute headquarter data to branches or collect branch data to headquarter, 1-to-N and N-to-1 sync are both supported.

- **Support Bidirectional Sync**

  The data in DB can be divided into multiple logic parts, each one can has it's own sync direction to achieve bidirectional sync in overall, this is useful for data sync in distributed system.

- **Support Binary Field**

  Some field type is binary, such as byte[], image or object, these fields are used for storing image files or binary data, they can synced in DBSync.

- **Nearly Real-time Sync**

  The minimum interval time for repeated sync can be 1 second. Once the source DB changes, the changed data will be synced to target DB in 1 second, the response is nearly real-time.

- **Unattended Sync**

  DBSync can run unattended, even thought the network is interrupted, or the computer is restarted, the sync process can restore automatically when fault cleaned, no data will lost.

- **Except Notification**

  DBSync can send except notification. Once error occurs it can send an Email to you so that you can handle the failure in time.

3 How It Works

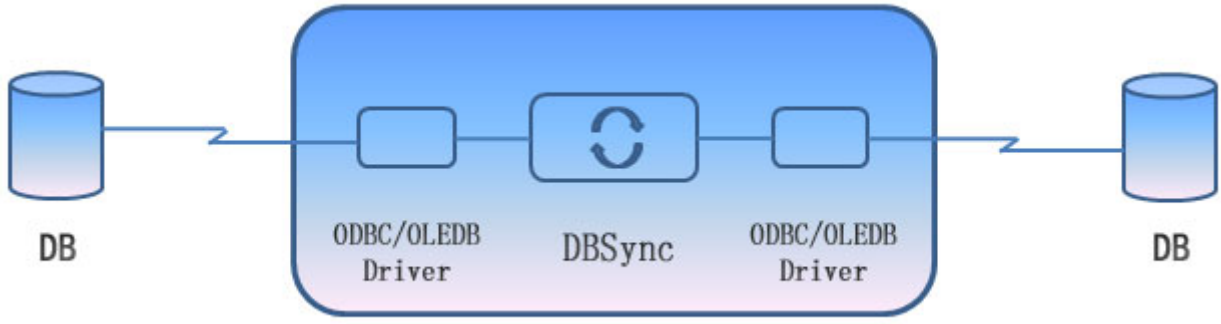

Figure 1：DBSync Mechanism

The mechanism of DBSync is simple and straight, it uses neither transaction logs nor triggers, it reads data from both sides, scans and compares them, finds out the differences, then writes the incremental data into target to keep both identical. There are two work modes:

The first is the simple scan mode, DBSync will scan and compare whole table data for each sync, it is suitable for small table (less than 100 million records) and low real-time sync.

The second is the incremental scan mode, DBSync can remember the finish time for each sync, and scan only the incremental data in the next sync. Since sync is done repeat, the increment between two syncs is little and the speed is fast. It is suitable for large table (more than 100 million records) and high real-time sync.

DBSync has no hand in internal processes of the database and only scans final data, this makes it widely applicable, highly reliable, and easy to use.

DBSync is designed for DB technicians, if you want to sync a DB, you should know its basic information, such as the DB location, the table structure, the user account and password etc.

4 How to Install

● Environment Required

DBSync is designed to run in windows system. You can install DBSync on the computer where source DB located or the computer where target DB located, or install it on the other computer where it can connect to both DB. Any computer that can connect to the database can run DBSync.

● How to Download

Visit Web Page: http://www.hc-software.com/DBSyncforeng.htm, then Click "Download" button.

● How to Install and Run

DBSync doesn't need to install, unzip DBSync.zip to a local directory, double click DBSync.exe to run.

Note: If your windows system has no .Net framework 2.0 installed, DBSync will pop up a window which direct you to install .Net Framework 3.5 (including. Net 2.0 and 3.0), do according to the prompt. If it works failed, you can also install manually, the steps are:
(1) Download the. Net 3.5 package from: http://www.hc-software.com/hcgis/support/NET Framework 3.5.zip
(2) Unzip to a temporary local directory, for example c:\tmp
(3) Right click the file net framework 3.5.bat → run as administrator → wait until the DOS window closed or say installation completed.

5 Basic Usage

5.1 Main Interface

After DBSync launches, the program will list all sync tasks:

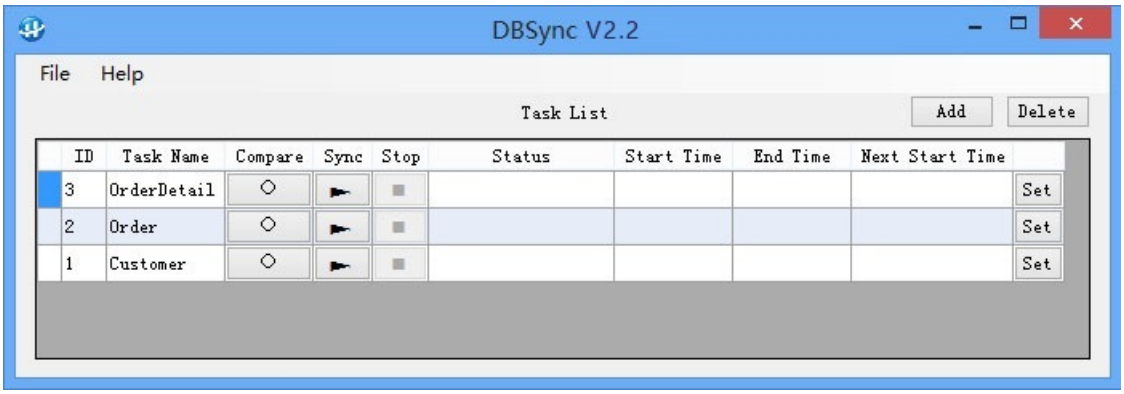

Figure 2：Main Interface

Note: Task is the basic unit of sync, one task synchronizes the data of a pair of tables. If there are more tables to sync, you need to set more tasks. There are 3 tasks by default, you can try and experience them directly.

The functions of buttons on the interface are:

- Add Task: click the "Add" button on the upper right corner to add a new task.
- Delete Task: click the "Delete" button on the upper right corner to delete a task.
- Compare Data: click the "○" button in task list to compare data.
- Sync Data: click the "►" button in task list to sync data.
- Stop Sync: click the "■" button to stop a running task.
- Status: show the progress information of a task, such as the row number has scanned, added, deleted, modified.
- Start Time, End Time, Next Start Time: show the time information of a running task.
- Set: click the "Set" button to modify a task.

5.2 Set Task

In order to compare and sync data, you need to set a sync task, in which you can set the database connect string, select table, define field mapping, select sync style and frequency, etc.

Step1: Connect Source and Target Database

Click the "Add" button on the top right of the main interface to create a task and enter page Step 1.

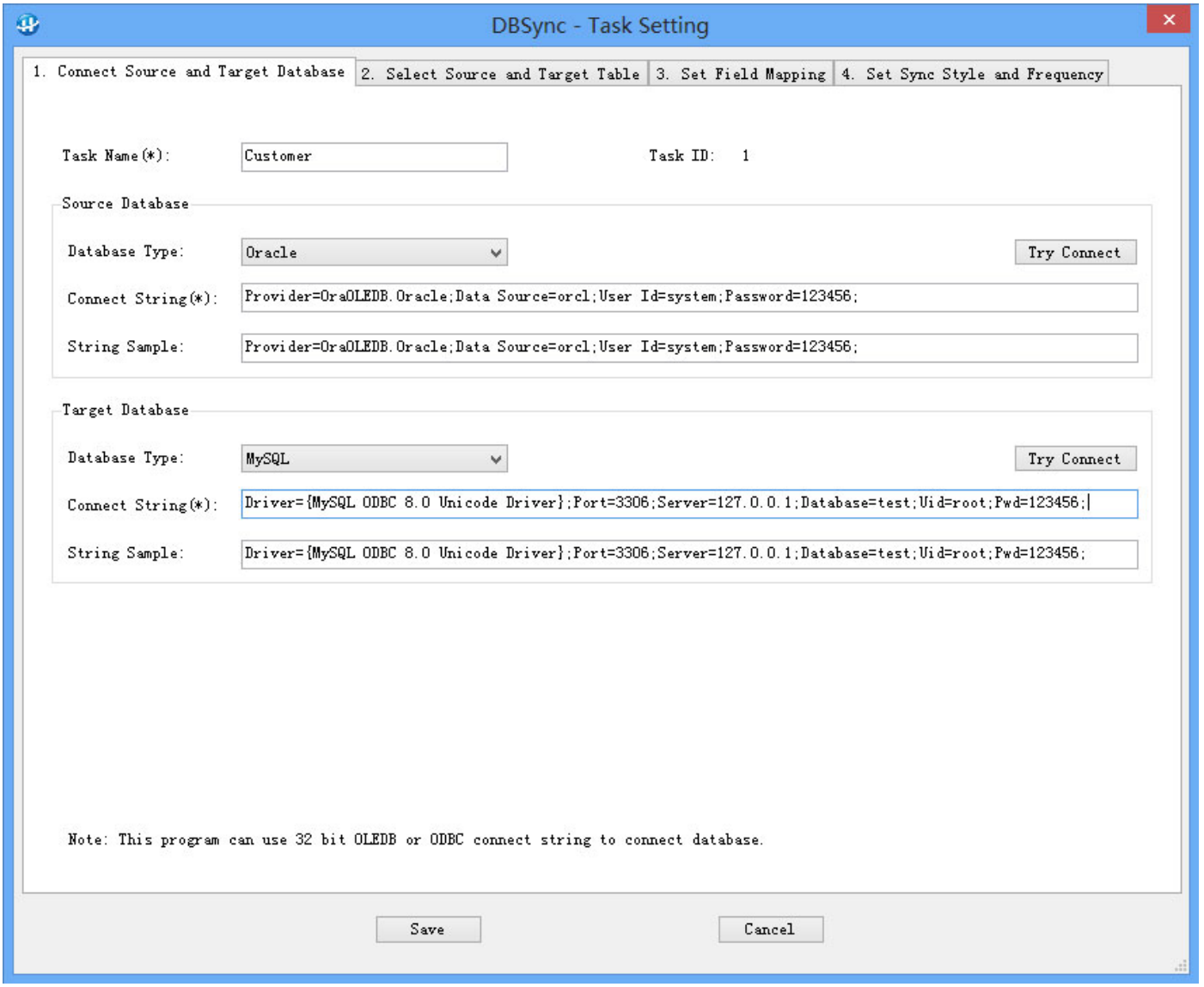

Figure 3：Connect to Source and Target DB

Note: DBSync uses driver to connect to the database, any data that can be connected with OLEDB or ODBC can be synchronized, so it supports a variety of databases, including SQL, NoSQL, data files, etc. This interface is used for setting database connection strings in order that DBSync can connect to DB, If the connection fails, please refer to Chapter 7.1 "How to Debug Database Connection" for detailed debugging steps and instructions.

Step2：Select Source and Target Table

Click Tab "2" on Task Setting interface to enter table select page.

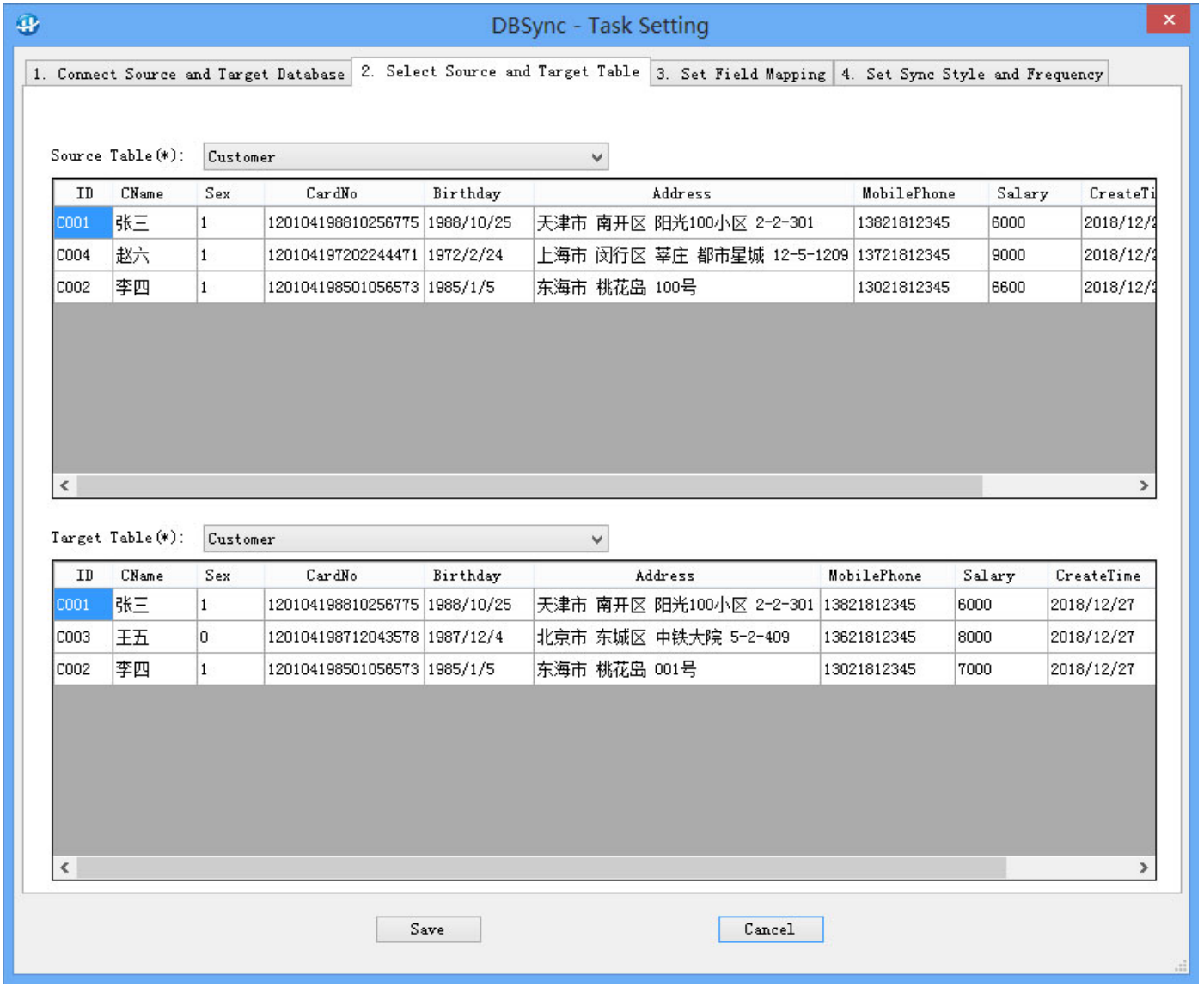

Figure 4：Select Source and Target Table

Note: This interface is used for selecting source table and target table. If there are too many tables in DB to find and select, you can enter the table name in the drop-down box to filter them. For example, if you enter character ‘abc’, only tables starting with ‘abc’ are listed; Or, if the database has schema mechanism, you can designate the schema in connection string, for example schema=hr will, only tables of schema hr are listed. Both methods can filter tables and make it easier to select.

Step3：Set Field Mapping

Click Tab "3" on Task Setting interface to enter field mapping page.

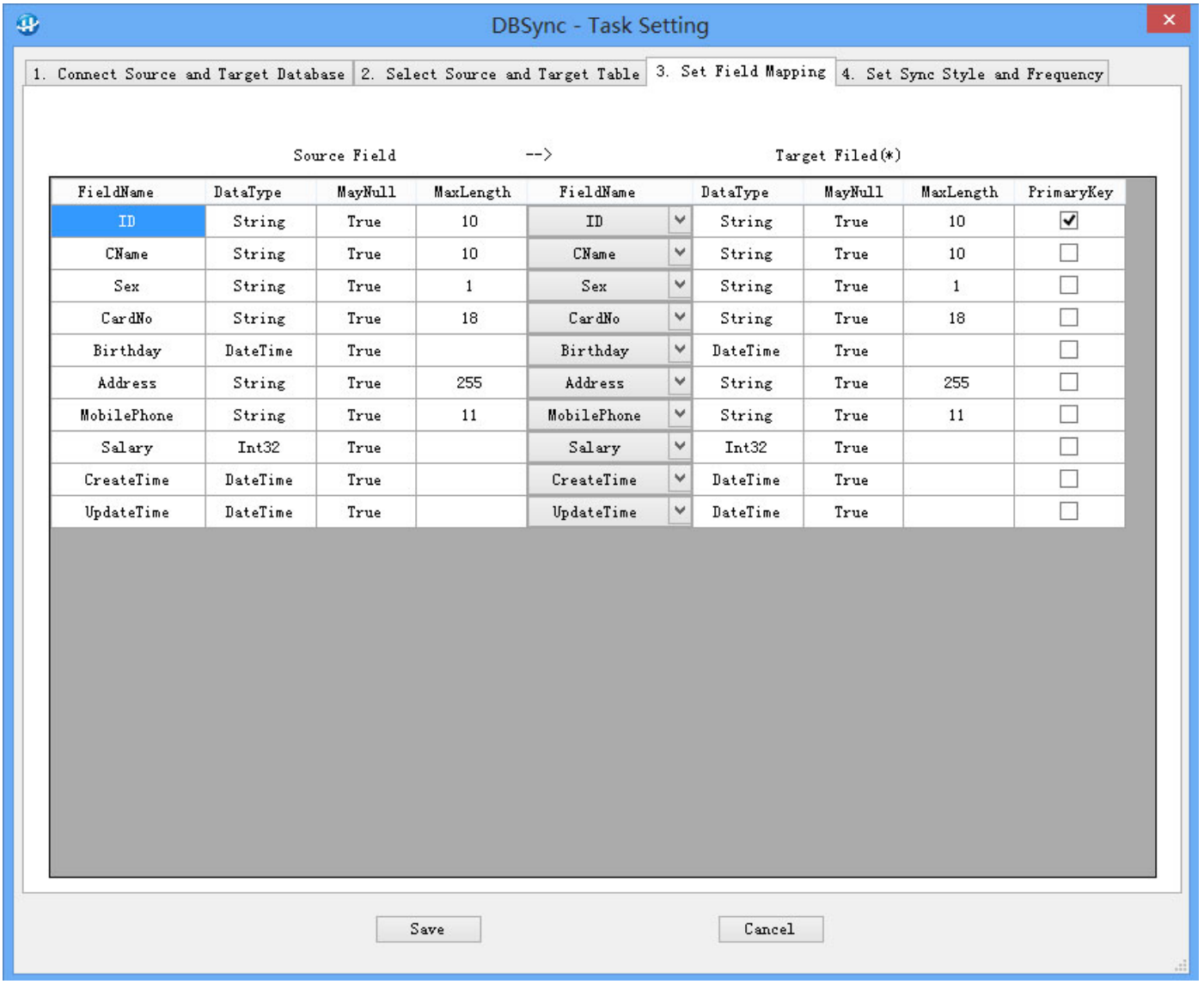

Figure 5：Set Field Mapping

Note: This interface is used for setting field mapping, if you don’t want to sync a source field, do not select target field for it. If you need Sync Differences, you must select primary key field to identify records.

Step4：Set Sync Style and Frequency

Click Tab "4" on Task Setting interface to enter sync style and frequency page.

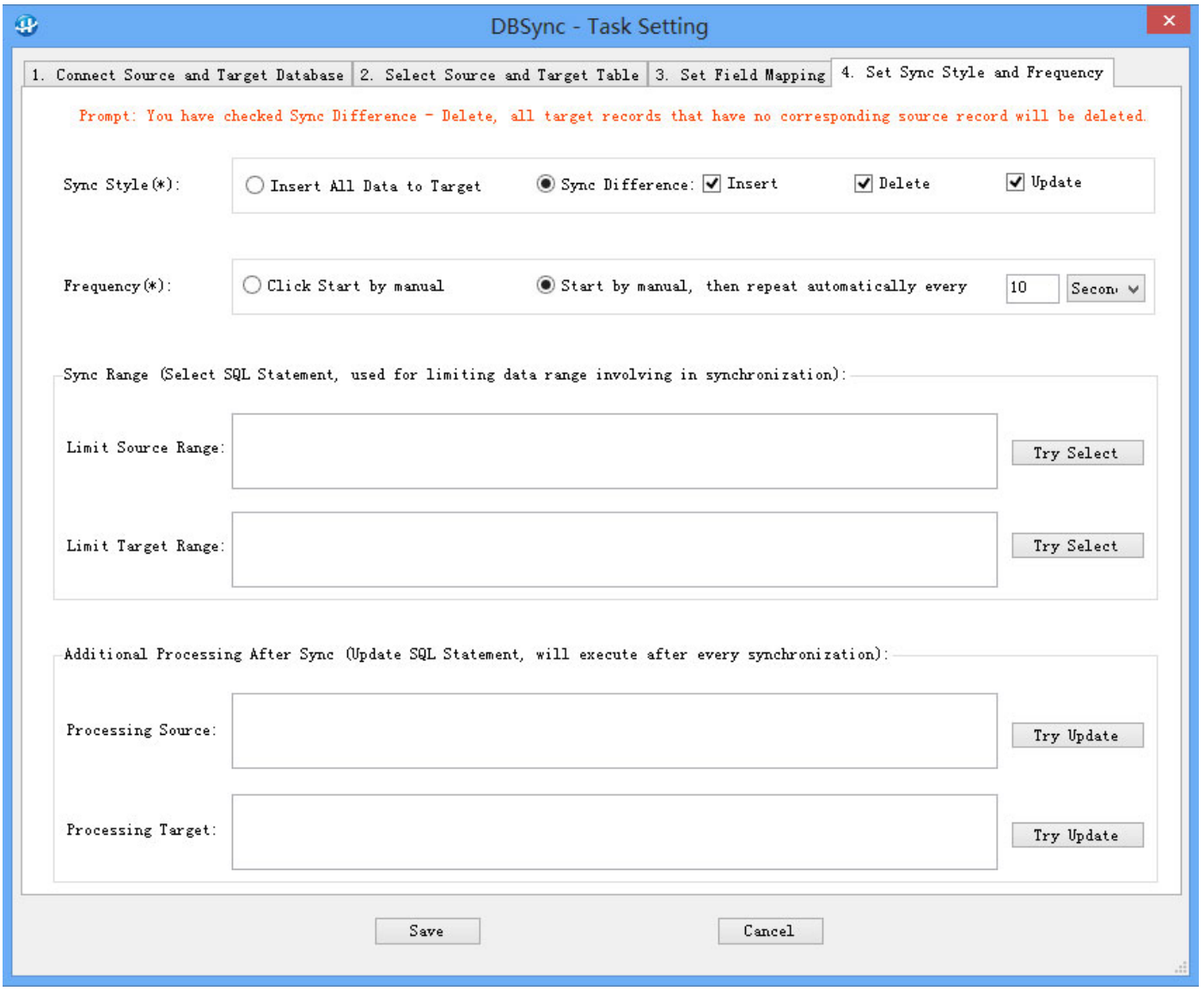

Figure 6： Set Sync Style and Frequency

Note: This interface is used for setting sync style, sync frequency, and other advanced options.

(1) Sync Style

■ Insert All Data to Target: Insert all data in the source table into the target table, this equal to copy data.

■ Sync Differences: Sync only the incremental data into the target. DBSync will use the primary key field as the record ID, find out the differences between two tables and sync only the differences. There are three types of difference:

A. Added records: The records exist in source table but don't exist in target table, they are regarded as added records in source, so they should be added to the target too.

B. Deleted records: The records don't exist in source table but exist in target table, they are regarded as deleted in source, so they should be deleted from target too.

C. Modified records: The records exist in both source table and target table, but there are some different field value, these records are regarded as modified in source, so they should be updated to the target.

Notes: The primary key field value should be globally unique and not null, otherwise thus records will be skipped and not be processed.

(2) Frequency

■ Click Start by Manual: Manually sync by clicking the "►" button in task list.

■ Start by Manual, then repeat automatically: Manually sync by clicking the "►" button in task list, when it is finished DBSync will sync repeatedly, as long as the program is not closed, the sync will go on.

(3) Sync Range

Advanced options, Normally they are not necessary to fill in except two situation:

First, if you need to limit data range involved in sync, you can fill in a SQL statement like select...where …to limit data rows.

Second, if the table name or field name conflicts with database reserved word, you can specify it clearly in SQL statement to avoid conflicts. For example, a table's name in your DB is Order, but this word is a reserved word, SQL statement select * from order will fail to execute. So you need to specify that it is a table clearly with a quote symbol: select * from 'order'. Each database has it's own symbol to clarify, Access and SQL Server use middle parentheses, Oracle uses double quote, MySQL uses reverse quote, etc.

Note: If the SQL statement is not select * but has specified field name, you should write down all fields, the number and order should be same as those specified in Step3, and the primary key fields should be listed at head.

(4) Additional Processing After Sync

Advanced options, Normally they are not necessary to fill in except this situation: If you need to perform additional data processing after syncing, for example data conversion, formula calculation, etc., you can fill in an update SQL statement in it, DBSync will execute it after every sync.

After all steps are completed, click save button to save the settings, a new task will be added to the task list.

5.3 How To Speed Up Task Setting

If there are many tables to sync, setting task one by one is a bit of boring and inconvenient. However, if source DB and target DB have same structure, DBSync has a mechanism to speed up this work, it can help you with two things:

First, when you create a task, if the field names are the same, the field mapping page (Step3) will automatically set by name. For example, if the fields of source table and target table are both F1, F2, F3, F4…, DBSync will automatically set to source F1 map to target F1, source F2 map to target F2,

source F3 map to target F3, etc.

Second, when you create a task, if the table names are the same, DBSync will automatically step forward by table names. For example, if the tables of source DB and target DB are both T1, T2, T3, T4, when you finish setting for T1, the next task will automatically turn to T2, and each page will try to use those of T1 as much as possible, including connection string, sync style, sync frequency, etc. After you finish setting for T2, the next task will automatically turn to T3, and each page will try to use those of T2... and so on. Therefore, if there are many tables need to sync, please set task in alphabetical order of table name to make use of this automation mechanism.

This function is also available between different type of database, for example, it is available too if the source is Oracle and the target is MySQL.

In short, as long as the table name or field name can be matched, DBSync will automatically match for you to help you speed up the setting process. However, if the names are not matched, you have to set it manually.

5.4 Data Compare Interface

After setting sync task, click the "○" button in task list to enter the compare interface:

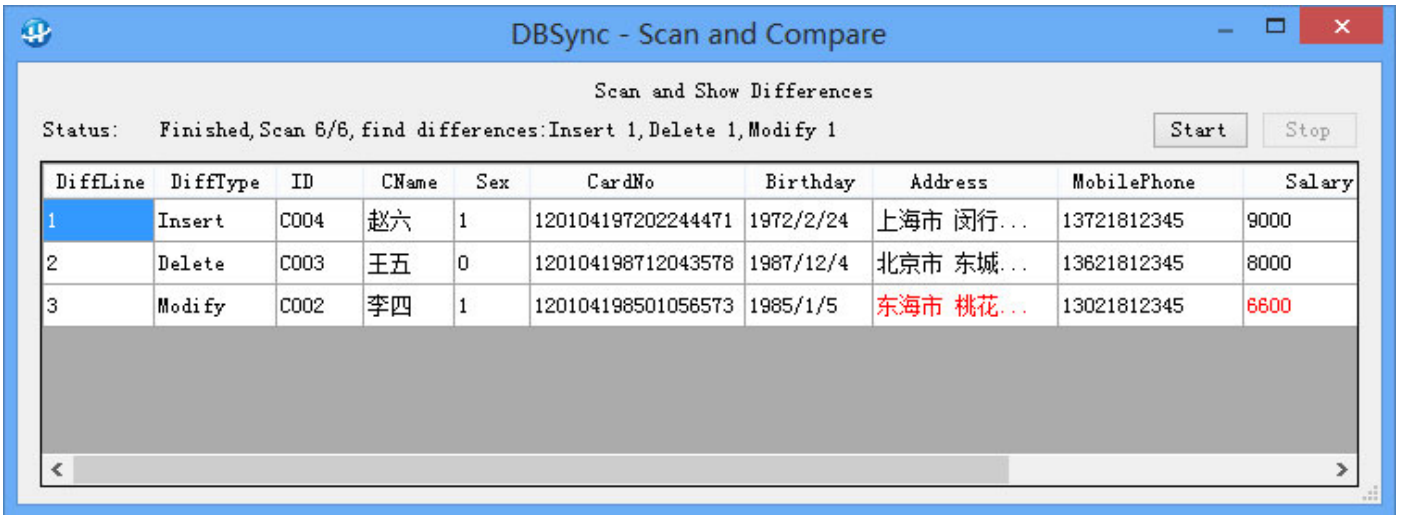

Figure 7：Data Compare Page

Note: This interface is used for comparing data, it will find and show the differences between two tables. The field value marked in red indicates that change has been made, move the mouse on it, the original value will display.

6 Advanced Usage

6.1 Part Sync

In some cases, we don't need to sync all data in a table, we just need to sync part of it.

There are two dimensions to limit the sync range. One is to limit the rows of the table, you can do it by setting Sync Range on Step4. For example, a company has a Sales Management System, in which the data in table Order has three states: draft, signed and paid. These order data need to sync to a Financial System, but only paid orders are required. So, you can fill in a SQL statement to limit the data range in source side: select * from order where status = 3 (status = 3 indicate paid orders). This task will sync only paid orders, other orders will not be synchronized.

Another dimension is to limit the fields of the table, you can do it by selecting target fields on Step 3. If the target field in the map line is blank, the source fields will not be synchronized.

6.2 Bidirectional Sync

What's bidirectional sync? Suppose there are two databases A and B, they both have tables T1, T2, T3 and T4. While tables T1 and T2 need to sync from A to B, T3 and T4 need to sync at a opposite direction B to A. This is a kind of bidirectional sync of database overall. Similarly, if a table can be divided into several blocks such as Part1, part2, part3 and Part4, while Part1 and part2 need to sync from A to B, Part3 and Part4 need to sync from B to A, this is bidirectional sync of table overall.

For example, the data in table Order is maintained by two systems, the sales information in it is entered by Sales System, the payment information in it is entered by the Financial System. This is essentially a distributed processing system. In order to sync the table, you need to divide the table by block and set sync tasks respectively. We can set task 1 for sales information and sync from Sales System to Financial System, then set task 2 for payment information to sync on opposite direction.

Therefore, the way of implementing bidirectional sync is to set sync tasks for different blocks respectively, each task has own direction. So, DBSync can meet the requirement of data sync for distributed system.

6.3 Sync Between Headquarter And Branches

Some company has a headquarter DB and several branch DBs. The headquarter has the entire data, while the branch has part of them. In order to sync data between them, we should limit the Sync Range on headquarter to make both sides are equal on data scope.

For example, the headquarter has customer data in all region, while a branch has only local customer data (for example Tianjin City). To sync between them two, you can fill in a SQL statement to limit

the Sync Range on headquarter side like this: select * from customer where area = Tianjin ", so that it is equal in data scope, this can avoid mixing other region's data. Similarly, if the headquarter side needs to sync with Beijing Branch, we can set another task and limit the data range to Beijing.

Anyway, we can always divide a headquarter database into several parts, each part sync with a branch database, this can realize the sync of 1 VS N or N VS 1.

6.4 View synchronization

Sometimes, the source data you want to synchronize is exposed in a view, but task setting cannot select view as source table, then how to sync? There is a workaround we could use. First, on the Step2 page of task setting, select a table that has the same or similar field structure as the view, such as Tabel1, and use it to finish the field mapping of Step3; then fill in a select statement in the Limit Source Range field of Step4, for example Select * from View1. DBSync will read source data from Views1 instead of Table1, to achieve view synchronization.

Note: If the SQL statement is not select * but has specified field name, you should write down all fields, the number and order should be same as those specified in Step3, and the primary key fields should be listed at head.

6.5 Dynamic database name and table name

In some cases, the database is not single and fixed, but is continuously generated over time, and there will be many databases. For example, in field data collection, attendance punching, log recording, there are often multiple data files, named by date or serial number. For this kind of data synchronization, the point to database cannot be fixed, it should be pushed forward and changed over time.

DBSync supports this kind of synchronization. It allows you embed Javascript or VBScript expressions in DB connection string. The expression can be calculated and replaced by running value to generate dynamic database name and DB pointing.

For example, in a field collection system, the data is stored in Access DB, every day there is a new database file generated, for example 20240313.mdb, 20240314.mdb, 20240315.mdb etc. In order to sync these data, you can write the file name:[vbscript:year(now) & right("0"&month(now),2) & right("0"&day(now)-1,2)].mdb, this VBScript expression can be calculated and replaced by running value, so we can have a dynamic pointing to database. Similarly, if the database is named by a serial number, you can calculate the interval value between two dates to make dynamic file name.

In the same way, if table name is not fixed and contains dynamic text such as date, serial number, etc., you can write a Select statement in the Limit Data Range in step 4, and embed a Javascript or VBScript expression to make dynamic table name.

For example, in a company's database, the name of its customer table is not fixed but a dynamic name with dates: Customer20240314、Customer20240315、Customer20240316…， You can write it like this: Select * from Customer[vbscript:year(now) & right("0"&month(now),2) & right("0"&day(now)-1,2)], it will be calculated and replaced based on the current date.

It should be noted that script expressions must be enclosed in square brackets and begin with Javascript: or VBScript:. If it is Javascript, you can write multiple statements separated by English semicolons, the calculation result is the value of the last statement, for example: [javascript:var a = new Date();b=a.getYear().toString ()+(a.getMonth()+1).toString();]. If it is VBScript, there can only be single one, for example: [vbscript:cstr(year(now)) & cstr(month (now))].

6.6 How To Speed Up

A key feature about DB sync is the performance, there are many factors that can affect the sync speed. The most important is the amount of data need to sync and the processing logic it use.

If Sync Style is Insert All Data to Target, the speed depends on the performance of target database and the amount of data to insert. The more data to insert, the longer time it will take, there is not much to do for improvement.

If Sync Style is Sync Difference, the program needs to scan and compare the data of both sides to find out the differences and then sync them. By default DBSync scan all data (Full Scan) to indentify differences, it’s straight and simple. But if the amount of rows in table is very large, for example there are more 100 million rows, the speed will slow down. On this case, you can set task to scan only the changed data (Part Scan). Since the sync is performed repeatedly, there will be little amount of changed data to scan every time, the speed will be very fast.

In order to perform part scan, the data in the table must have time information such as create time and update time, then you can set task like this:
■For added and modified data. Set a task and fill in a SQL statement at Sync Range on Step4 to limit the scan range for source DB. For example, if you want to sync once every 10 minutes, you can scan only the data inserted or modified in the past 10 minutes. As for as SQL server, the SQL statement should be:
select * from t where datediff(n,createtime,getdate())<=10 or datediff (n,updatetime,getdate())<=10.
If the table has no create time and update time fields, you can add a timestamp field to the table, whenever a record is added or modified, the field will be assigned the latest timestamp value automatically, so we can use it as the time fields.
■For deleted data, set another task separately. The task will not involve the modified data, it compare only the primary keys, the speed is very fast even though the data range is not limited.

6.7 Auto Restore Sync

When DBSync close and start, the tasks that were running before closed can restore automatically,

no manual operation is required. Base on this feature, we can configure it to restart whenever computer reset to prevent accidental stop of sync process. There are two ways to configure:

One way is to add DBSync to the startup directory and make it run in foreground when computer reset. The steps is: Start → all programs → startup → right click mouse → resource manager, enter the startup program directory → right click mouse in the blank space → new → shortcut → browse → select the DBSync.exe. After that, whenever computer reset, DBSync will run automatically in foreground, and you can see the software interface and running status.

Another way is to convert DBSync to windows service and make it run in background when computer reset. The steps is: Enter DBSync directory → Enter Service subdirectory → double click DBSyncservice.bat, this will add a Windows Service named DBSync, whenever computer reset DBSync service will run automatically in background. Because windows service has no interface, there is not interface displayed.

If you want to run DBSync as a service, there are some thing to pay attention to. First, do not run it in foreground and background at the same time to avoid conflicts. Second, the background mode is only available for full version of DBSync, not available for free version, because the free version can run only in the foreground. The third is the order of launching it, you should set all sync tasks and start them first, then close the program, and then launch the background service. If you need to modify task setting, you should stop the service first, then return to the foreground to modify, and then launch the background service.

You can uninstall the service when you don't need it, the steps is: stop the service, enter the service subdirectory, double click removeDBSyncservice.bat, the service will be removed.

6.8 Data Security

DBSync is a secure software and has multiple security options.

Firstly, it runs independently on local machine and does not connect or exchange data with the outside, so it will not leak data.

Secondly, it need neither to set triggers, stored procedures, scripts in the database, nor to modify database options, so it will not affect database system.

Thirdly, DBSync uses a connection string to connect to DB, you can use account in the string to limit access rights For example, for the source database, use a read-only account, there will be not possibility to modify it; For the target database, use a restrict accounts that can only write the tables required to sync, there will be not possibility to modify other tables.

Fourthly, for remote synchronization, DBSync can adopt a secure connection, and data transmission will be encrypted to prevent information leakage. For example, for an ODBC connection to MS SQL Server, adding Encrypt=Yes to the connection string and configuring a security certificate on the

database side will use SSL encryption for data transmission. Different drivers have different encryption transmission settings, please refer to their respective technical materials or use the driver configuration interface to see if they support it. For example, in the configuration interface of SQL Server's ODBC driver, the option "use strong encryption for data" indicates encrypted transmission, check the box as following figure:

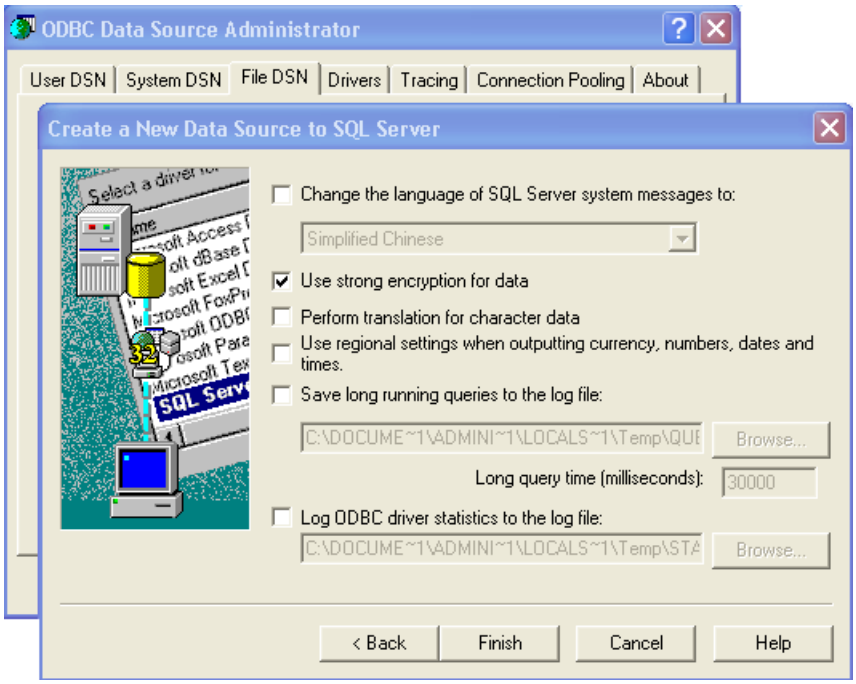

Figure 8：SQL Server ODBC Configuration Interface

Note: If the driver supports encrypted connections, how can we write the connection string? You can create a new file DSN, configure and save it, and then open the file with Notepad to see the option grammar and add it to the database connection string.

6.9 Set Email Notification

The software can send email notification when task is finished, so that you can know the result of the task, this is especially useful when some error occurs during sync. There are three kinds of sync results. The first is that sync is finished with no error. The second is that sync is finished but there is some error occurs, for example, some data has problem and sync cannot be done for these data. The third is that sync is interrupted, for example, the network is interrupted and sync has to be paused.

Note: If you set task to repeat automatically, it will try sync at a regular time interval despite of the previous failure. Occasional network interruption and database failure affect only the current sync action, not affect the next time, once the problem is solved the sync will resume and continue, so DBSync is very stable and firm.

You can define how to notify in file DBSync.ini:
[MailNotify]
NotifySuccess=0

NotifyError=1
NotifyInterrupt=1
MailServer=smtp.qq.com
LoginUID=xxxx@qq.com
LoginPWD=123456
MailFrom= xxxx@qq.com
MailTo= xxxx@qq.com
MailCC=

The meaning of parameter is:
NotifySuccess: Whether send notice when task is finished with no error. 1:send, others:doesn't send.
NotifyError: Whether send notice when task is finished with errors. 1:send, others:doesn't send.
NotifyInterrupt: Whether send notice when task is interrupted. 1:send, others:doesn't send.
MailServer: The address or IP of email server.
LoginUID: Sender's login account.
LoginPWD: Sender's login password.
MailFrom: Sender's email address.
MailTo: Receiver's email address, if there are multiple receivers, please separate them with semicolons.
MailCC: CC Receiver's email address, if there are multiple receivers, please separate them with semicolons.

The definition in the [MailNotify] paragraph applys to all tasks. If some task need a different setting, you can set an individual paragraph for it. The paragraph should name [TaskXXMailNotify], where XXX is the task number. For example, if task23 doesn't need any notification, you can add a paragraph like this:
[Task23MailNotify]
NotifySuccess=0
NotifyError=0
NotifyInterrupt=0

7 References

7.1 How to Debug Database Connection String

DBSync uses connection string to connect to database, it provides sample string for each kind of database, so the easiest way of getting connection string is: Select database type first, the sample string will appear on the interface, replace the IP address, user name, password in it with your own, you will get your own connection string. Click the button Try Connect, it should prompt "Connection success", if it fails, please debug as following steps.

Step1: Using client tool to connect

First, put aside DBSync and use database client tool to connect to see if database works fine. Each

database has it's own client application, such as SQL Developer for Oracle, Workbench for MySQL, Management Studio for SQL server, pgAdmin4 for PostgreSQL, etc; Besides, third-party tools is available too, such as Navicat, DataGrid, DBeaver, etc.

If the client tools cannot connect, there may be some problem with the database, or the network doesn't work, or the connection port is not open, please ask DBA for help. If it can connect, it means the database works fine, please go to Step 2.

Step2: Using driver program to connect

If database client tool can connect to DB, you can use driver program to connect to see if driver works fine.

DBSync can use OLE DB driver or ODBC driver. Which one it actually uses is decided by the connection string you fill in. If the connection string starts with Provider=xxx, the OLE DB driver is used, if it starts with Driver=xxx, the ODBC driver is used, where xxx is the driver name.

The steps of using driver to connect DB is as follows:

- OLE DB driver

If you are running in 32-bit Windows systems, double-click DBSync.udl in the DBSync directory to launch the DB Link Properties program. Click the Provider tab first, then select a provider that suits you, and then click Next to set DB connection. If you are running in 64-bit Windows systems, double clicking on. udl will launch a 64 bit driver program, which is not suitable for DBSync. You need to start a 32-bit driver program, the steps are: Start → Run → Enter this command: Rundll32.exe "C:\Program Files (x86)\Common Files\System\Ole DB\oledb32.dll", OpenDSLFile d:\dbsync\DBSync.udl → Enter. (Here we assumed DBSync is located at d:\dbsync, if it is not so, please modify the path as yours.)

- ODBC driver

If you are running in 32-bit Windows systems, the steps are: Start → Control Panel → System and Security → Administrative Tools → Data Sources (ODBC) → Double click it to launch the ODBC Data Source Manager. Click the File DSN tab, then click Add button, then select a driver that suits you, and then click next to set DB connection. If you are running in 64-bit Windows systems, please note that there are two ODBC data sources(32-bit and 64 bit), please use 32-bit instead of 64 bit. If 32-bit ODBC is not listed in the control panel, launch it like this: Start → Run → Enter c: \ windows \ syswow64 \ odbcad32.exe → Enter.

If the driver specified by connection string cannot be found, the driver maybe not installed. Please download and install it according to Chapter 7.2. Some databases have multiple drivers, if one doesn't work, try another, if none of them work, please contact the official database support. If it can connect, it means the driver works fine, please go to Step 3.

Step3: Correct the connection string

If driver program can connect, DBSync do can connect. The only reason why it connects fail is that there is spelling error in connection string. For example, the driver name specified in connection string does not match the name in Step 2, or the login account & password are wrong, please check and correct it. If it still connects fail, you can turn to use the driver's own spelling way to make connection string, do as follows:

Firstly, you need to know where the configuration parameters in Step 2 are saved. For OLE DB drivers, the parameters are saved in the DBSync.udl you double-click on; For ODBC drivers, the parameters are saved in the DSN file you created.

Next, open the file with Notepad to see the connection properties and spelling way. Concatenate them together and separate them with semicolons to make connection string. Copy and paste it into the DBSync to connect. It should be noted that the password entered in Step 2 will not store in file, so you need to write it manually in the format of PWD=123456 or Password=123456.

After the above efforts, if you still cannot connect to DB, please contact DBSync official support.

7.2 Connection String Examples and Driver

Here we list 21 kinds of usually used database connection string and their driver location for download. As for other rarely used databases, please refer to their technical documents, as long as they can be connected, data sync is supported.

■ Oracle

OLEDB connection string:
Provider= OraOLEDB.Oracle; Data Source=orcl;User Id=sa;Password=123456;
Note: The definition of orcl such as host, port, service name, etc., should be defined in tnsnames.ora.

ODBC connection string:
DRIVER={Oracle in OraDb11g_home1};DBQ=orcl;Uid=sa;Pwd=123456;
Note: The definition of orcl such as host, port, service name, etc., should be defined in tnsnames.ora.

Driver download: https://www.oracle.com/database/technologies/dotnet-odacdev-downloads.html, please download the latest 32-bit ODA OUI, such as ODA 18.3. During installation, there are many component options in step 4, only Oracle Provider For OLEDB is required, the others are not.

■ MySQL

MySQL does not provide official OLE DB Provider, we can only use ODBC driver.

ODBC connection string:

Driver={MySQL ODBC 8.0 ANSI Driver};Server=127.0.0.1;Port=3306;Database=test;Uid=root;Pwd=123456;

Driver download: https://dev.mysql.com/downloads/connector/odbc, Detail steps: On the download page, select Microsoft Windows for Operating System, select Version till Windows (x86, 32-bit) MSI Installer appear, then click Download on the right. During installation, if you are prompted that visual studio x86 redistributable is missing, please download and install it first from this page: http://www.hc-software.com/hcgis/support/vc_redist(2015-2019).x86.exe, then install MySQL ODBC driver.

■ SQL Server 2000

OLEDB connection string:
Provider=sqloledb;Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

OLEDB trusted connection string:
Provider=sqloledb;Server=(local);Database=Northwind;Integrated Security=SSPI;

ODBC connection string:
Driver={SQL Server};Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

ODBC trusted connection string:
Driver={SQL Server};Server=(local);Database=Northwind;Integrated Security=SSPI;

Driver download: You don’t need to do that, these drivers are included in windows system by default.

■ SQL Server 2005

OLEDB connection string:
Provider=SQLNCLI;Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

OLEDB trusted connection string:
Provider=SQLNCLI;Server=(local);Database=Northwind;Integrated Security=SSPI;

ODBC connection string:
Driver={SQL Native Client};Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

ODBC trusted connection string:
Driver={SQL Native Client};Server=(local);Database=Northwind;Integrated Security=SSPI;

Driver download: Install SQL Server 2005 client application, these drivers will be available.

■ SQL Server 2008

OLEDB connection string:
Provider=SQLNCLI10;Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

OLEDB trusted connection string:
Provider=SQLNCLI10;Server=(local);Database=Northwind;Integrated Security=SSPI;

ODBC connection string:
Driver={SQL Server Native Client 10.0};Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

ODBC trusted connection string:
Driver={SQL Server Native Client 10.0};Server=(local);Database=Northwind;Integrated Security=SSPI;

Driver download: Install SQL Server 2008 client application, these drivers will be available.

■ SQL Server 2012\2014\2016\2017\2019\2022

OLEDB connection string:
Provider=SQLNCLI11;Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

OLEDB trusted connection string:
Provider=SQLNCLI11;Server=(local);Database=Northwind;Integrated Security=SSPI;

ODBC connection string:
Driver={SQL Server Native Client 11.0};Server=(local);Database=Northwind;Uid=sa;Pwd=123456;

ODBC trusted connection string:
Driver={SQL Server Native Client 11.0};Server=(local);Database=Northwind;Integrated Security=SSPI;

Driver download: Install SQL Server client application, these drivers will be available.

■ SQL Server CE

SQL Server CE is the abbreviation of SQL Server Compact Edition, it has no official ODBC driver, we have to use OLEDB connection, and it can only be used as source database, not target database.

OLEDB connection string:
Provider=Microsoft.SQLSERVER.CE.OLEDB.4.0;Data Source=c:\mydb.sdf;

Runtime and driver download: https://www.microsoft.com/en-us/download/details.aspx?id=30709, please download and install the 32-bit package.

■ PostgreSQL

PostgreSQL has no official OLE DB provider, please use ODBC connection.

ODBC connection string:
Driver={PostgreSQL Unicode};Port=5432;Server=127.0.0.1;Database=test;Uid=postgres;Pwd=123456;

Driver download: https://www.postgresql.org/ftp/odbc/versions/msi, please download and install the 32-bit package.

■MongoDB

MongoDB is a document-based NoSQL database, its Collection and Document are map to relational DB's table and row. DBSync supports data sync between MongoDB and MongoDB, as well as MongoDB and relational DB. Document and row can be converted to each other.

As for official driver, ES provides only ODBC driver, the driver can read MongoDB but can not write MongoDB, so MongoDB can only used as sync source, cannot used as sync target. Besides, the server side is required to install MongoDB Connector for Bi.

Official ODBC connection string:
DRIVER={MongoDB ODBC 1.4.2 Unicode Driver};Server=127.0.0.1;Port=3307;UID=;PWD=;Database=test;

Official ODBC Driver download: https://github.com/mongodb/mongo-bi-connector-odbc-driver/releases
Official Mongo-BI-Connector download: https://www.mongodb.com/zh-cn/products/bi-connector

As for third-party driver, CDATA provides ODBC driver, it can read and write MongoDB, so it can used as both sync source and sync target. You can try this driver for free for one month, after that you need to pay for it for long-term usage.

CData ODBC connection string:
DRIVER={CData ODBC Driver for MongoDB};Server=127.0.0.1;Port=27017;Database=test;
User=;Password=;

CData ODBC Driver download: https://www.cdata.com/drivers/mongodb/odbc

■ DB2

OLEDB connection string:
Provider=DB2OLEDB;Network Transport Library=TCPIP;Network Address=127.0.0.1;Initial

Catalog=MyCtlg;Package Collection=MyPkgCol;Default Schema=Schema;User ID=sa;Password=123456;

ODBC connection string:
driver={IBM DB2 ODBC DRIVER};hostname=127.0.0.1;Database=test;protocol=TCPIP; uid=sa;pwd=123456;

Driver download: https://www.ibm.com/support/fixcentral, select Information Management as the Product Group, then select IBM Data Server Client Packages, then select an appropriate Runtime Client Package.

■Elasticsearch (ES)

Elasticsearch (ES) is a search service based on document storage, its indices, index and document are map to relational DB's database, table and row. DBSync supports data sync between ES and ES, as well as ES and relational DB. Document and row can be converted to each other.

As for official driver, ES provides only ODBC driver, the driver can read ES but can not write ES, so ES can only used as sync source, can not used as sync target. Besides, the server side is required to have Platinum License.

Official ODBC connection string:
Driver={Elasticsearch Driver}; server=127.0.0.1; port=9200; UID=; PWD=; secure=0;

Official ODBC Driver download: https://www.elastic.co/cn/downloads/odbc-client, select MSI 32-bit as the Platform, then click download.

As for third-party driver, CDATA provides ODBC driver, it can read and write ES, so ES can used as both sync source and sync target. You can try this driver for free for one month, after that you need to pay for it for long-term usage.

CData ODBC connection string:
DRIVER={CData ODBC Driver for Elasticsearch};Server=127.0.0.1;Port=9200; User=; Password=;

CData ODBC Driver download: https://www.cdata.com/drivers/elasticsearch/odbc

■ SQLite

SQLite has neither officially OLE DB provider nor ODBC driver, we have to use the third party ODBC driver.

ODBC connection string:
Driver= Driver={SQLite3 ODBC Driver};Database= c:\myDB.db;

Driver download: http://www.hc-software.com/hcgis/support/sqliteodbc.zip

■ Firebird

Firebird has no official OLE DB provider, please use ODBC connection.

ODBC connection string:
Driver={Firebird/InterBase(r) driver};Uid=SYSDBA;Pwd=123456;DbName= c:\mydb.fdb;

Driver download: http://firebirdsql.org/en/odbc-driver, please download and install the latest 32-bit package.

■ Access（.mdb）

Standard OLEDB connection string:
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\mydb.mdb;Persist Security Info=False;

OLEDB connection string with password:
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\mydb.mdb;Persist Security Info=False; ;Jet OLEDB:Database Password=123456;

Standard ODBC connection string:
Driver={Microsoft Access Driver (*.mdb)};Dbq= c:\mydb.mdb;

ODBC connection string with password:
Driver={Microsoft Access Driver (*.mdb)};Dbq= c:\mydb.mdb;Pwd=123456;

Driver download: You don’t need to do that, these drivers are included in windows system by default.

■ Access（.accdb）

Standard OLEDB connection string:
Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\mydb.accdb;Persist Security Info=False;

OLEDB connection string with password:
Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\mydb.accdb;Persist Security Info=False; ;Jet OLEDB:Database Password=123456;

Standard ODBC connection string:
Driver={Microsoft Access Driver (*.mdb, *.accdb)};Dbq= c:\mydb.mdb;

ODBC connection string with password:
Driver={Microsoft Access Driver (*.mdb, *.accdb)};Dbq= c:\mydb.mdb;Pwd=123456;

Driver download: Install office 2007 or above version, these drivers will be available.

■ Excel

OLEDB connection string:
Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\myExcel.xls;Extended Properties="Excel 8.0;HDR=YES";
Note:
A. When you access excel file by database interface, the excel file will act as database, the sheet in Excel will act as table.
B. HDR = yes indicates the first row in Excel is the header line.

ODBC connection string:
Driver={Microsoft Excel Driver (*.xls)};Dbq= c:\myExcel.xls;
Driver download: You don't need to do that, this driver is included in windows system by default.

■Text File

OLEDB connection string:
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=c:\myPath;Extended Properties="text;HDR=Yes;FMT=Delimited";
Note:
A. When you access txt\csv file by database interface, the file directory will act as database, the file in directory will act as table, so the Data Source in connect string should be the directory path.
B. HDR = yes indicates the first row in file is the header line.
C. FMT=Delimited indicates the fields is separated by comma.

ODBC connection string:
Driver={Microsoft Text Driver (*.txt; *.csv)};Dbq= c:\myPath;Extensions=asc,csv,tab,txt;

Besides, if a file is used as the target database, some sync operations are not supported, they are:
Sync to TXT, CSV and other text files: insert is supported, delete and update are not is supported.
Sync to Excel file: insert and update are supported, delete is not supported.

Driver download: You don't need to do that, this driver is included in windows system by default.

8 Technical Support

Web Site: http://www.hc-software.com/DBSyncforeng.htm

Telephone: 0086-22-28211389
Email: support@hc-software.com