

# DBSync

# User Manual

V2.0

2024.3

1	WHAT' S DBSYNC.....	2
2	MAIN FEATURES.....	2
3	HOW IT WORKS.....	3
4	HOW TO INSTALL.....	4
5	BASIC USAGE.....	5
	5.1 MAIN INTERFACE.....	5
	5.2 SET TASK.....	5
	5.3 HOW TO SPEED UP TASK SETTING.....	10
	5.4 DATA COMPARE INTERFACE.....	11
6	ADVANCED USAGE.....	12
	6.1 PART SYNC.....	12
	6.2 BIDIRECTIONAL SYNC.....	12
	6.3 SYNC BETWEEN HEADQUARTER AND BRANCHES.....	12
	6.4 VIEW SYNCHRONIZATION.....	13
	6.5 DYNAMIC DATABASE NAME AND TABLE NAME.....	13
	6.6 HOW TO SPEED UP.....	14
	6.7 AUTO RESTORE SYNC.....	15
	6.8 SET EMAIL NOTIFICATION.....	15
7	REFERENCES.....	17
	7.1 HOW TO DEBUG DATABASE CONNECTION STRING.....	17
	7.2 CONNECTION STRING EXAMPLES AND DRIVER.....	18
8	TECHNICAL SUPPORT.....	25

## 1 What's DBSync

DBSync is a lightweight desktop software used for database synchronization, it supports a variety of DB, it supports incremental sync, heterogeneous sync, remote sync, and provides a permanent free version.

## 2 Main Features

### ● Direct and Simple to Use

It's very direct and simple to use DBSync. You don't even need to install it, just download it and unzip it to use.

### ● Support a Variety of Databases

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

### ● Support Heterogeneous Sync

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

### ● Support Incremental Sync

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

### ● Data Filtering and Processing

You can limit data range involved in sync task, the range can be whole DB or some tables in DB or some rows & fields in table, you can filter them as you like.

### ● 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 part can has it's own sync direction to achieve bidirectional sync in overall, this is useful for the sync for distributed system.

### ● Support Binary Field

Some field hold binary type, such as byte[], image or object field, they are used for storing image files or binary data, sync of these fields are supported 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 can 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 the failure restored, no data will lost.

### ● Except Notification

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

## 3 How It Works

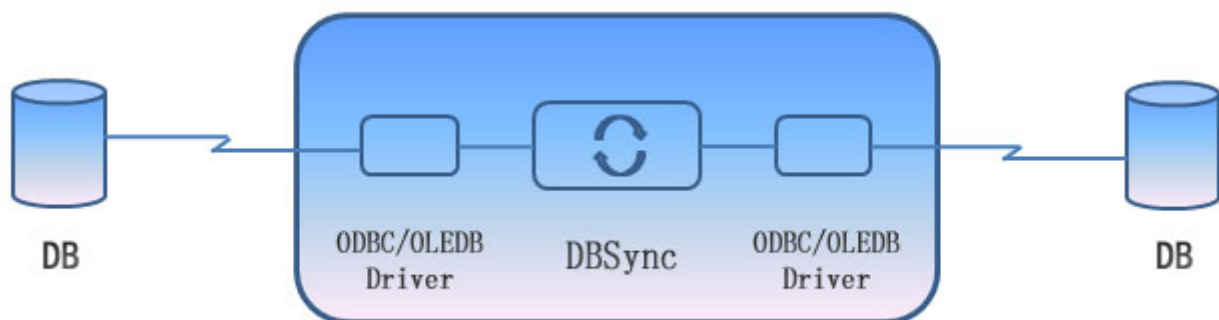


Figure 1: DBSync Mechanism

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 incremental data into target to keep both identical. There are two working modes:

The first is the simple scan mode, which scans and compares the entire table for each sync, which is suitable for small volume of data (less than 100 million levels) and low real-time requirements.

The second is the incremental scan mode. It can remember the time for each sync, and scan only the incremental data in the next sync. Since sync is performed regularly, the increment between two syncs is very small and the speed is very fast. It is suitable for large volume of data (more than 100 million levels) and high real-time requirements.

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

DBSync is designed for DB technicians, if you want to sync DB data you should know the basic information about the DB. For example, in order to sync customer data, you should know its location and structure, you should have access permission for it.

## 4 How to Install

### ● Environment Required

DBSync is designed to run in windows system. You can install DBSync on the side of source DB or the side of target DB, or install it on a third-party computer. Any computer that can connect to the database can install 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, the program 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 it manually, the steps are:

(1) Download the .Net 3.5 package from: <http://www.hc-software.com/hcgis/support/NETFramework3.5.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 it prompt the installation completed.

## 5 Basic Usage

### 5.1 Main Interface

When DBSync startup, the program will list the sync tasks:

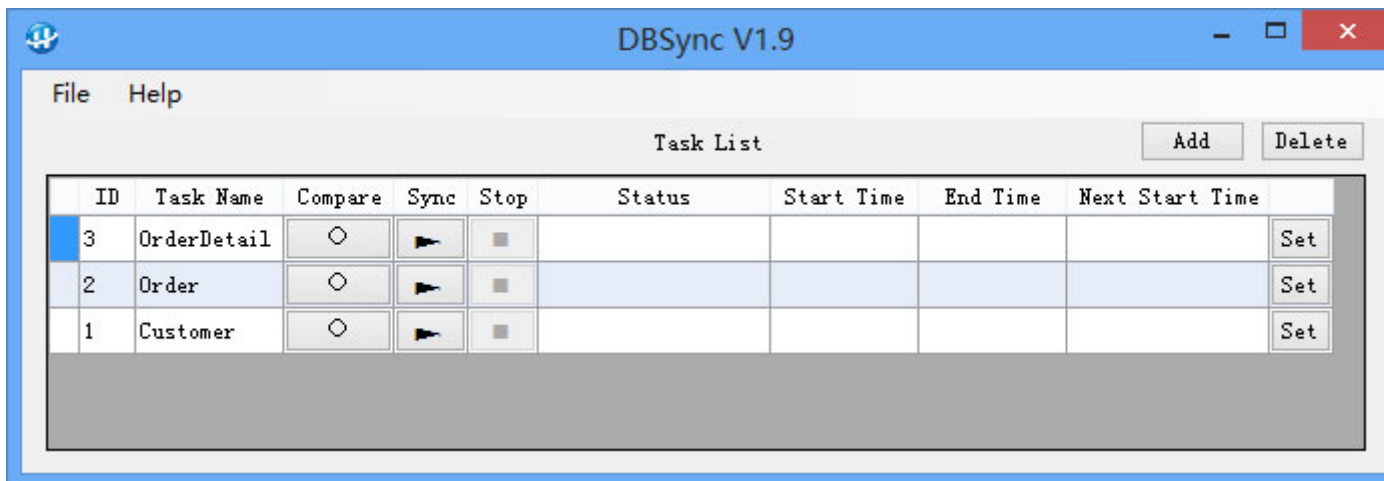


Figure 2: Main Interface

Note: Task is the basic unit of sync, one task synchronizes the data of a pair of tables. If you want to sync data of more tables, you need to set more tasks. There are 3 tasks by default, you can execute 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 to start sync data.
- Stop Sync: click the "■" button to stop a running task.
- Status: show sync progress, such as the row number task has scanned, added, deleted, modified.
- Start Time, End Time, Next Start Time: display the time information of a running task.
- Set: click the "Set" button to modify the settings of a task.

### 5.2 Set Task

In order to compare and sync data, you need to set sync task, in which you can set the database connect string, select table, define field mapping, sync style and sync frequency, etc. Click the "Add" button on the top right of the main interface to add a new task and enter page Step 1.

#### Step1: Connect Source and Target Database

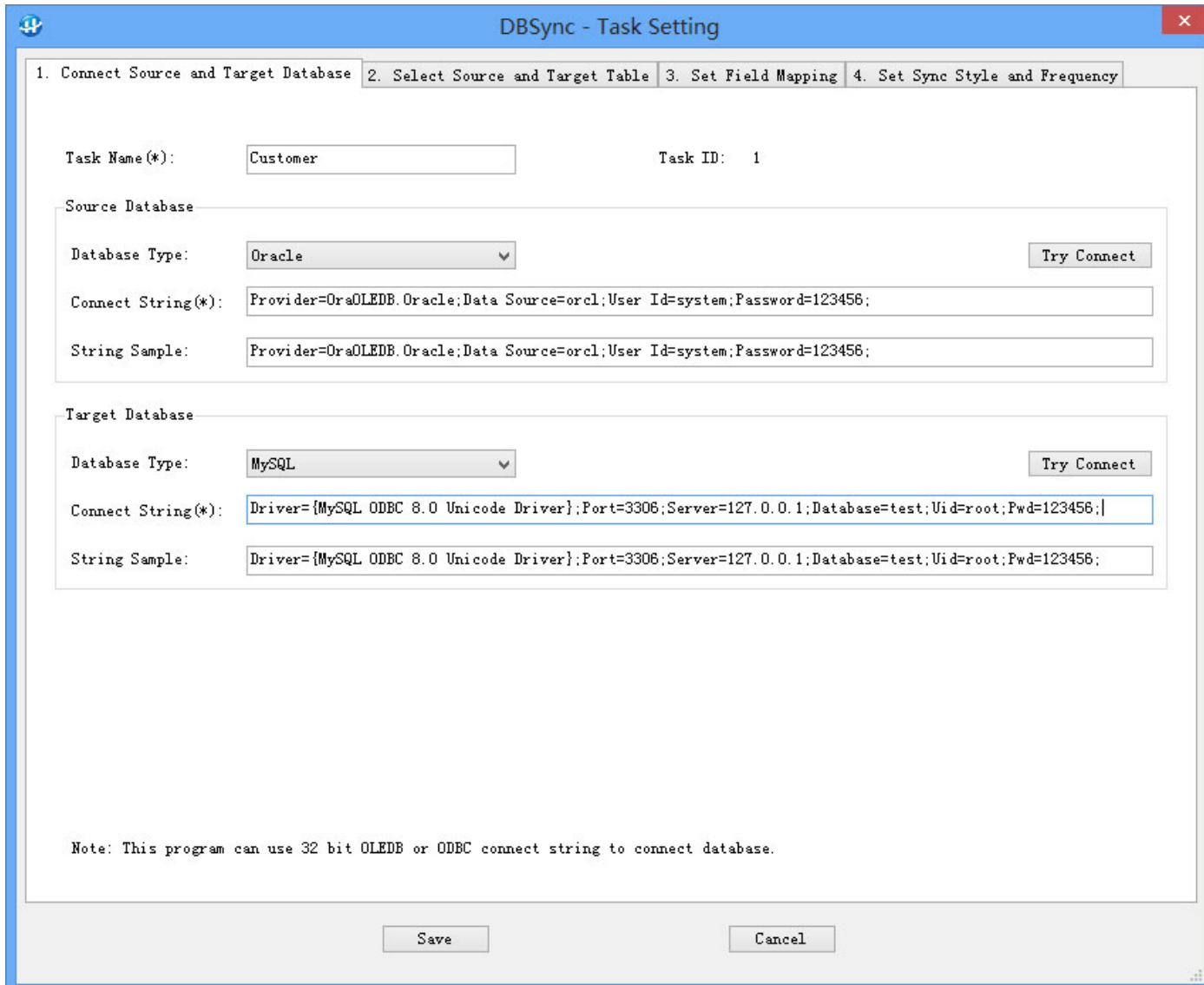


Figure 3: Connect to Source and Target DB

Note: DBSync uses a driver to connect to the database, any data that can be connected with OLEDB or ODBC can be synchronized, so it supports various databases, including SQL, NoSQL, data files, etc. This interface is used for setting database connection strings, you can fill OLEDB or ODBC connect string here, the connect driver should be 32 bit.

After setting connection strings, click button Try Connect, if it prompt "Connect Success", click tab 2 to enter Step 2. If connect fails, please see chapter 7.1 "How to Debug Database Connection String" where will tell you the detail troubleshooting steps.

## Step2: Select Source and Target Table

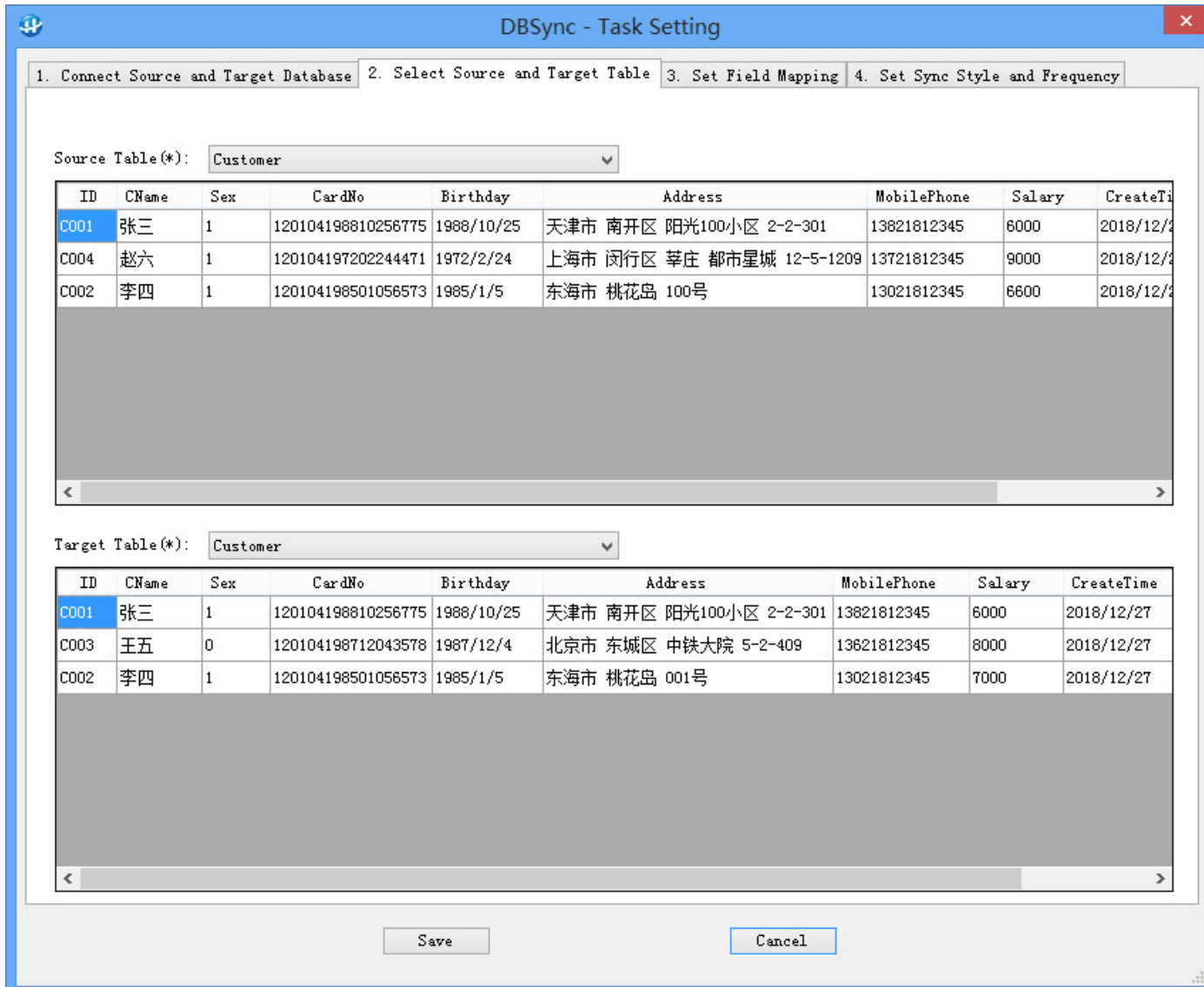


Figure 4: Select Source and Target Table

Note: This interface is used for selecting source table and target table.

### Step3: Set Field Mapping

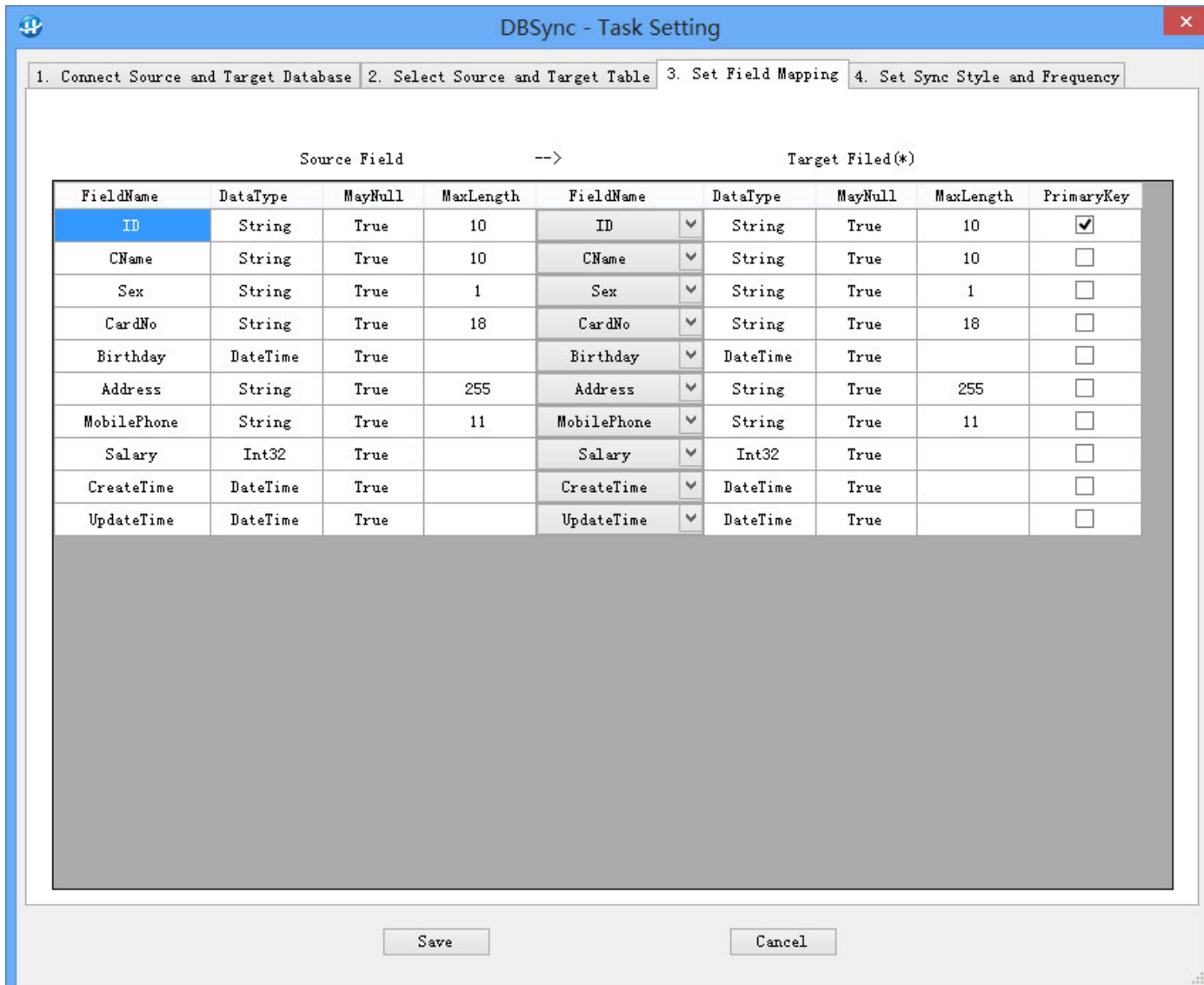
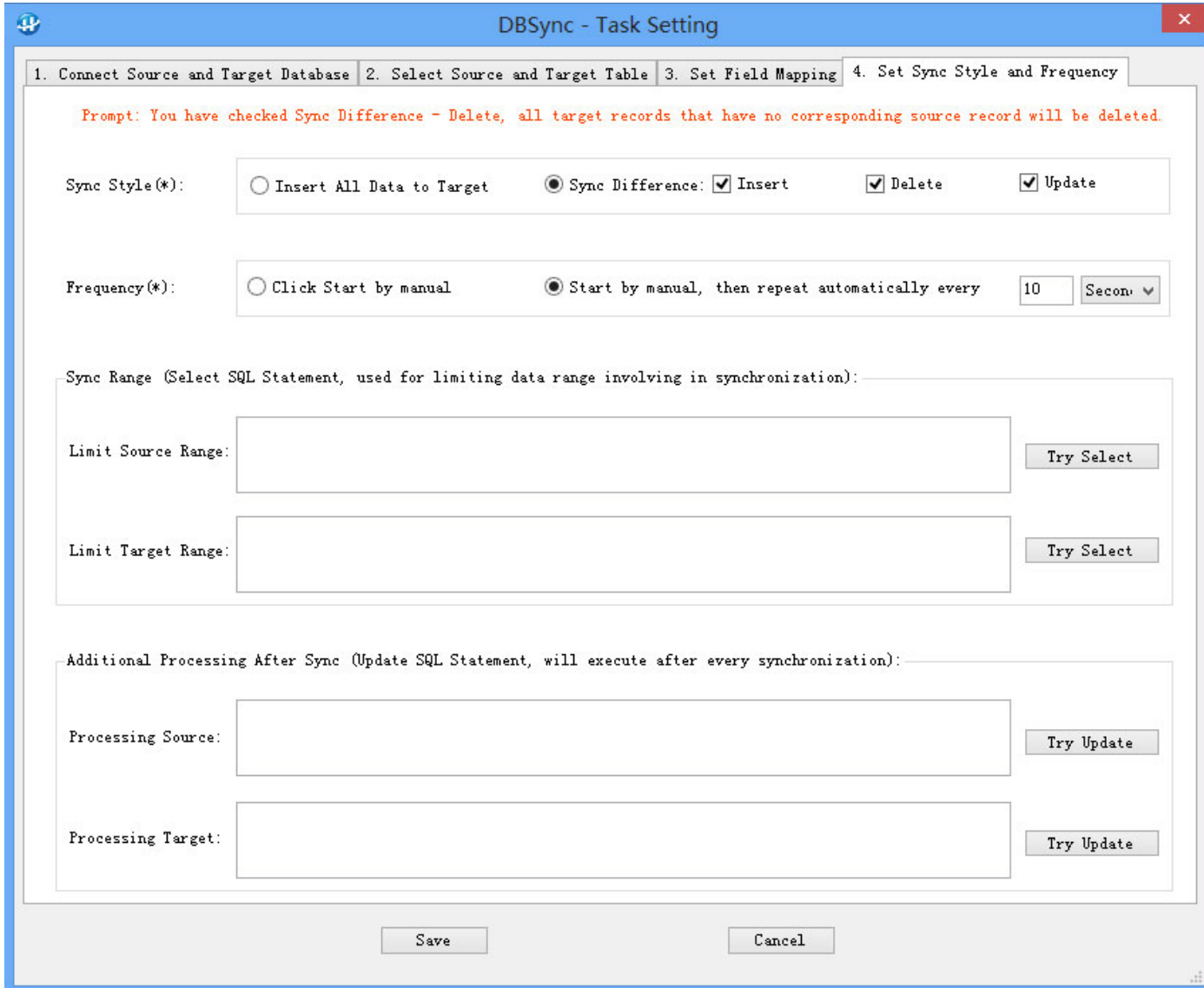


Figure 5: Set Field Mapping

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 check primary key field to identify each record.

**Step4: Set Sync Style and Frequency**





DBSync - Task Setting

1. Connect Source and Target Database 2. Select Source and Target Table 3. Set Field Mapping 4. Set Sync Style and Frequency

Prompt: You have checked Sync Difference - Delete, all target records that have no corresponding source record will be deleted.

Sync Style(\*):  Insert All Data to Target  Sync Difference:  Insert  Delete  Update

Frequency(\*):  Click Start by manual  Start by manual, then repeat automatically every

Sync Range (Select SQL Statement, used for limiting data range involving in synchronization):

Limit Source Range:

Limit Target Range:

Additional Processing After Sync (Update SQL Statement, will execute after every synchronization):

Processing Source:

Processing Target:

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 is equivalent to copy data.
- **Sync Differences:** Just sync the incremental data into the target. DBSync will use the primary key field as the record ID, search for 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 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 some fields are different, they 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 shouldn't be null, records that have null value or duplicated value will be skipped and will not be processed.

## (2) Frequency

- Click Start by Manual: Manually perform sync by clicking the "▶" button in task list.
- Start by Manual, then repeat automatically: Manually perform sync by clicking the "▶" button in task list, when sync process 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 name is Order, but the word is reserved word of the database, SQL statement `select * from order` will fail to execute. So you need to specify it is a table clearly with a quote symbol: `select * from 'order'`. Each database has its 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 *` and you have specified field name, you should write down all fields, the number and list order of fields should be consistent with those specified in Step3, and the primary key fields should be listed at first.

## (4) Additional Processing After Sync

Advanced options, Normally this is not necessary to fill in except this situation:

If you need to perform additional data processing after syncing, such as 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, task setting one by one is a bit of boring and inconvenient. However, if the source DB and the target DB have same structure, DBSync has a mechanism to speed up the process of task setting. There are two things DBSync automatically does for you:

First, when you add a new 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 add a new 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.

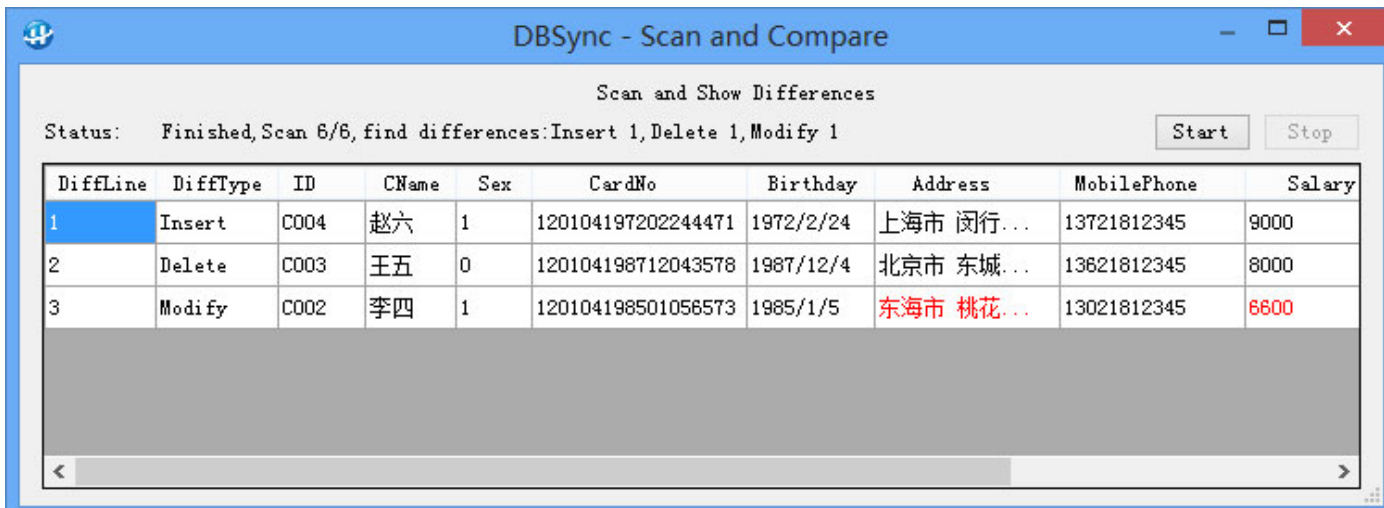
The automation mechanism is also available between different type of database, for example, it is available if the source is Oracle and the target is MySQL.

In short, as long as the table name and field name can be matched, DBSync will automatically match for you to help you speed up the setting process.

However, if the database structures are different and the names are not matched, you have to set manually one by one.

## 5.4 Data Compare Interface

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



The screenshot shows a window titled "DBSync - Scan and Compare" with a sub-header "Scan and Show Differences". The status bar indicates "Status: Finished, Scan 6/6, find differences: Insert 1, Delete 1, Modify 1". There are "Start" and "Stop" buttons. The main area contains a table with the following data:

DiffLine	DiffType	ID	CName	Sex	CardNo	Birthday	Address	MobilePhone	Salary
1	Insert	C004	赵六	1	120104197202244471	1972/2/24	上海市 闵行...	13721812345	9000
2	Delete	C003	王五	0	120104198712043578	1987/12/4	北京市 东城...	13621812345	8000
3	Modify	C002	李四	1	120104198501056573	1985/1/5	东海市 桃花...	13021812345	8600

Figure 7: Data Compare Page

Note: This interface is used for comparing the data of both sides, it will search and show the differences. The field value marked in red indicates 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 them.

There are two dimensions to limit the sync range. One is to limit the rows of the table, you can get 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 get 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. Tables T1 and T2 need to sync from A to B, while T3 and T4 need to sync at a opposite direction of B to A. This is a kind of bidirectional sync of database overall. Similarly, if a table is divided into several blocks such as Part1, part2, part3 and Part4, Part1 and part2 need to sync from A to B, while Part3 and Part4 need to sync from B to A, this is bidirectional sync of table overall. As for the method of block division, you can divide by fields or rows.

For example, table Order is maintained by two systems respectively. The sales information in it is entered by Sales System, while the payment information in it is entered by the Financial System. This is essentially a distributed processing system. To sync all information, 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 DB. The headquarter has full data, the branch database has part of them. To sync between them, we should limit the Sync Range on headquarter side 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 apply a SQL statement to limit the Sync Range on headquarter side like this: `select * from customer where area = Tianjin "`, so that they are equal in data scope, this can avoid the mixing in of 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 will 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 located in a view, and the task settings cannot use the view as the source table, so how to synchronize it? There is actually a workaround here. First, on the Step2 page of task settings, select a table with the same or similar field structure as the view, such as Table1, and use it to complete 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 be read source data from Views1 instead of Table1, thereby achieving view synchronization.

It should be noted that if the Select statement you fill in is not `Select *`, but fills in specific field names, then it must be filled in completely. The number and order of fields should be consistent with the fields you specified in Step 3, and the primary key Fields are sorted first. In other words, if you don't fill it in, it's fine. If you fill it in, it must be completely consistent.

## 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, serial number, etc. For this kind of data synchronization, the database pointing cannot be fixed, but must be continuously pushed forward over time and dynamically pointed to each database.

DBSync supports this kind of synchronization. It allows Javascript or VBScript expressions to be embedded in the database name of the data connection string. The expression can be calculated in real time and the result is replaced, thereby dynamically generating the database name and realizing dynamic pointing.

For example, in a field collection system, the data is stored in Access, and one database file is generated every day, such as `Data20240313.mdb`, `Data20240314.mdb`, `Data20240315.mdb`...etc. To synchronize these data, you can write the file name like this: `Data[vbscript:year(now) & right("0"&month(now),2) & right("0"&day(now)-1,2)].mdb`, the VBScript expression in it can be calculated in real time based on the current date. For example, today is 2024-3-15, the calculation result is 20240314, replacing it will get the file name `Data20240314.mdb`, and you can synchronize

the database; turn to tomorrow, it will be calculated and point to Data20240315.mdb, and turn to the day after tomorrow it will point to Data20240316.mdb... and so on, thereby achieving dynamic pointing and synchronization of the database. Similarly, if the database is named by a serial number, you can get a numerical number by subtracting the date in the expression. When the date is pushed forward, you can get a serial number. In short, the advantage of script expressions is very flexible.

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

For example, a company's database has a table Customer. Normally, the SQL statement should be written like this: `Select * from Customer where xxx`. But if the table name is not fixed and is a dynamic table name containing date, you can write it like this: `Select * from Customer[vbscript:year(now) & right("0"&month(now),2) & right("0" &day(now)-1,2)] where xxx`, it will be calculated and replaced based on the current date during synchronization. For example, today is 2024-3-15, and the actual SQL statement executed is `Select * from Customer20240314`; tomorrow, the actual executed SQL statement is `Select * from Customer20240315`...and so on.

It should be noted that script expressions must be enclosed in square brackets and must 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 one, and it must be an expression, for example: `[vbscript:cstr(year(now)) & cstr(month (now))]`.

## 6.6 How To Speed Up

A key feature of database sync is the performance, there are many factors can affect the speed of sync. The most important is the amount of data to sync and the processing logic you 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 no much thing 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. DBSync will scan all data (Full Scan) by default to indentify differences, the advantage is 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 a 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 that fill in a SQL statement at Sync Range on Step4 to

limit the scan range for source DB. For example, if your sync is repeatedly every 10 minutes, you can scan only the data inserted or modified in the past 10 minutes. For example in SQL server:  
`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, so that if a record is added or modified, the field will be assigned the latest timestamp value automatically, it can be used as time information.

■ For deleted data, set another task separately. Since 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 again, the executing tasks before will restore run automatically, no manual operation is required. Base on this feature, we can configure it restart after 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 detail 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 status information.

Another way is to convert DBSync to windows service and make it run in background when computer reset. The detail steps is: Enter the service subdirectory under the DBSync directory, double-click the file DBSyncservice.bat, it will add a Windows Service named DBSync. After that, whenever computer reset, DBSync service will run automatically in background. Because windows service has no interface, the status information are invisible.

When running as a background service, there are some thing you should 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 applicable to full version of DBSync, not available the free version, because the free version can operate only in the foreground. The third is the order of launch, you should set all sync tasks and start them first, and then close the program, then launch the background service. If you need to modify task setting, you should stop the service first, and then return to the foreground to modify, and then launch the background service.

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

## 6.8 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. The third is that sync is interrupted, for example, the network is interrupted and the database cannot be connected.

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 will affect only the current sync action, not affect the next time, when the failure is removed, 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 to send a notice when task is finished with no error. 1:send, others:doesn't send.

NotifyError: Whether to send a notice when task is finished with errors. 1:send, others:doesn't send.

NotifyInterrupt: Whether to send a notice when task is interrupted. 1:send, others:doesn't send.

MailServer: The address of the server used for sending email.

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 is applicable to all tasks by default. If some task has special request in notification, you can define 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 ensure database works well. Each database has its 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 client tool can connect, that means the database is ok, the problems should be the driver or the connection string, please enter Step2 for further debug. If client tool connect failed, please consult your DBA, maybe the database has shutdown, or the connect port is forbidden.

#### **Step2: If there is no driver, download and install it first**

If you get a prompt "XXXX provider is not registered on the local computer", or a prompt "data source name is not found and no default driver is specified", these indicate that your windows system is lack of database connection driver. Please download and install according to the Chapter 7.2.

Note: DBSync requires 32-bit driver. If the connection string starts with "driver=" DBSync will use ODBC drive, if it starts with "provider=" DBSync will use OLE DB driver. After driver installing, Try Connect in DBSync again, if it still fails, please go to Step3.

#### **Step3: Use the driver to connect and correct the connection string**

Any error in connection string will cause fails, and it is a bit of difficult to troubleshoot. We suggest you put aside DBSync and use the driver itself to connect, if it fails either, please consult database official for support. If it works well, learn how driver spell the connect string, and correct the error in your connection string, then try connect again. Do as follows:

##### ■ Get OLE DB connection string

If you are using OLE DB connection driver and 32-bit Windows systems, double-click DBSync.udl in the DBSync directory to open the tool Database Connection Property → click Provider → select a provider → next → configure connection parameters → test the connection → Confirm. Get connection string: DBSync.udl → right click → open method → Notepad → learn how it is spelled.

For 64 bit Windows systems, double-click DBSync.udl will pop-up the 64 bit tool, it is not applicable to DBSync. The steps of start the 32-bit tool program: start → run → enter command line: rundll32.exe "C:\Program Files (x86)\Common Files\System\Ole DB\oledb32.dll",OpenDSLFile d:\DBSync\DBSync. UDL → OK.

Note: Here we assume DBSync is located in d:\DBSync, please correct it by your own location.

#### ■ Get ODBC connection string

If you are using ODBC connection driver and 32-bit Windows systems, Click Start → control panel → system and security → management tools → data source (ODBC) → file DSN → add → select driver → next → configure connection parameters. Get connection string: use Notepad to open the file DSN you created, join the attribute values together and separate them with semicolons. If your database connection requires a password and the file DSN does not contain it, you need to add it manually. The format is: PWD=123456.

For 64 bit Windows systems, notice that there are 32-bit and 64 bit ODBC data sources, please use 32-bit instead of 64 bit, other steps are the same. If there is no 32-bit ODBC in the management tool, please start it manually. Steps: start → run → enter c:\windows\syswow64\odbcad32.exe

After correcting, if it still fail to connect, please consult DBSync technician for support.

## 7.2 Connection String Examples and Driver

Here we list 21 kinds of commonly used database connection string and their driver address. 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 ODAC. In step 4 of installation, there are many component options, please check Oracle Provider For OLEDB, other components are not required.

## ■ 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 Windows (x86,32bit) for OS Version, find Other Downloads at the bottom of the page, find MSI Installer below it, 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](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

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 when you use it to connect, 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. When you use it to connect DB, MongoDB 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 the 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 when you use it to connect, 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. When you use it to connect DB, 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 the 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 (.acldb)

Standard OLEDB connection string:

Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\mydb.acldb;Persist Security Info=False;

OLEDB connection string with password:

Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\mydb.acldb;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](mailto:support@hc-software.com)