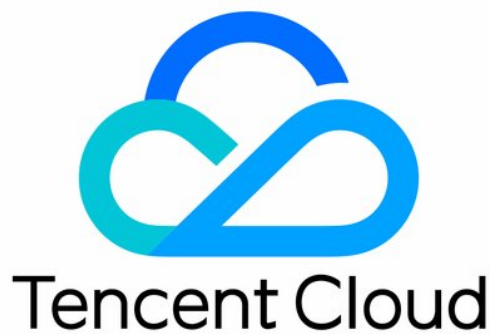


# Data Transmission Service

## FAQs

### Product Documentation



## Copyright Notice

©2013-2022 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## FAQs

### NewDTS

Data Migration

Data Sync

FAQs for Data Subscription Kafka Edition

Regular Expressions for Subscription

### OldDTS

General

General

# FAQs

## NewDTS

# Data Migration

Last updated : 2022-09-16 10:08:43

## General

### **Will the data in the source database be deleted after migration with DTS?**

No. Data migration with DTS essentially replicates the data from the source database to the target database without affecting such data.

### **How does data migration with DTS affect the target database?**

When data is migrated to the target database, the system will verify whether the source and target databases have tables with the same name, and if so, the verification will fail, and you will be prompted to make changes first.

### **Does DTS support migration from one offline database to another?**

No. DTS only supports migration from self-built databases, databases in other clouds, or TencentDB databases to TencentDB databases.

### **Does DTS support data migration between TencentDB instances under two different Tencent Cloud accounts?**

Yes. For migration between TencentDB instances under two different Tencent Cloud accounts, you need to log in to DTS with the Tencent Cloud account of the target instance.

### **Can I configure multiple DTS tasks for migration from the same source database to different TencentDB instances?**

Yes. You can migrate data from the same source database to multiple target databases and vice versa, but multiple concurrent tasks may increase the access pressure on the source and target databases and thus slow down the migration. If you need to create multiple migration tasks for the same source database, then after creating the first task, you can quickly create similar tasks by clicking **More > Create similar task** in the **Operation** column.

### **Does DTS support scheduled automatic migration?**

Yes. When modifying the configuration for a created data migration task, you can select scheduled migration and specify the start time.

## Can I monitor the task progress during migration?

Yes. You can log in to the [DTS console](#) and view the migration task progress on the **Data Migration** page.

## Why is there a 15-day limit on incremental migration?

Currently, incremental migration is performed through the nearest proxy server via Tencent Cloud Direct Connect, which eliminates network jitters and ensures the quality of data transfer. The 15-day limit can reduce the connection pressure on the proxy server and is only intended for reasonable utilization of resources for migration. Connections will not be force closed after 15 days.

## How is the data accuracy ensured during data migration?

DTS uses Tencent Cloud's proprietary data migration architecture to verify the data accuracy in real time and quickly detect and correct errors. This guarantees the reliability of the transferred data.

## Why does data verification require that the source database instance not be read-only?

This is because data verification requires creating a new database `__tencentdb__` in the source instance and writing the checksum table to the database. If the instance is read-only, data verification will be skipped.

## Can I specify tables for migration with DTS?

Yes. You can select the entire instance or specify tables as the migration object.

## When does data migration stop?

When you select incremental migration, if it takes a long time before the task stops, you may need to stop it by yourself.

- If you select **Structural migration** or **Full migration** as the **Migration Type**, the task will automatically stop upon completion.
- If you select **Full + Incremental migration** as the **Migration Type**, after full migration is completed, the migration task will automatically enter the incremental data sync stage, which will not stop automatically. You need to click **Done** to manually stop the incremental data sync.
  - Manually complete incremental data sync and business switchover at appropriate time.
  - Check whether the migration task is in the incremental sync stage without any lag. If so, stop writing data to the source database for a few minutes.
  - Manually complete incremental sync when the data gap between the target and the source databases is 0 MB and the time lag between them is 0 seconds.

## Why does the data size change before and after full migration?

This is because the fragmented spaces of the source and target databases are different, and the source database may contain data holes. In this case, after full migration is completed, the table storage space in the target database

may be smaller than that in the source database. We recommend you perform a data consistency check as instructed in [Creating Data Consistency Check Task](#) after the migration is completed to check whether the contents of the source database and the target database are consistent.

### Is double write supported during data migration?

No. Writing data to both the source and target databases during migration may cause data inconsistency.

### Does DTS support cross-region database migration?

Yes. You can implement cross-region data transfer over the public network.

## MySQL

### Will tables be locked during MySQL migration?

Data migration between MySQL, MariaDB, and TDSQL-C, as well as from TDSQL for MySQL to TDSQL for MySQL, MySQL, or MariaDB is performed without locks by default. This doesn't mean there are no locks at all; instead, there is merely no global lock (the FTWRL lock).

Migration from MySQL or MariaDB to TDSQL for MySQL is performed with locks by default.

- Migration without locks: Only tables without a primary key are locked.
- Migration with locks: A global lock (the FTWRL lock) will be applied for seconds and then released after the consistency offset is obtained.

### Are TencentDB for MySQL single-node (formerly the Basic Edition) instances supported for migration?

- A TencentDB for MySQL single-node (formerly the Basic Edition) instance can be used as the source database for migration over the public network, but not over the private network.
- A TencentDB for MySQL single-node (formerly the Basic Edition) instance cannot be used as the target database for migration.

### After setting `binlog_format` to `row` for the source database, how do I ensure that the `binlog_format` of the source database takes effect immediately?

After setting `binlog_format` to `row`, you need to reset all business connections to the current database. If the source database is a replica, you also need to reset the master/replica sync SQL thread to prevent current business connections from writing data in the old format.

Before the above operation is completed, do not create or start a migration task; otherwise, data inconsistency may occur.

**What should I pay attention to if the TokuDB engine is used in the source instance?**

In this case, TokuDB will be converted to InnoDB by default during migration. As tables containing clustered indices or compressed with TokuDB need to be preprocessed before migration, they are not supported currently. DDL operations on TokuDB are not supported either.

**Can I migrate user permissions during migration of MySQL data?**

Yes. NewDTS supports migrating user permissions.

# Data Sync

Last updated : 2022-10-19 18:25:38

## **Will the sync service be affected if high availability (HA) switch occurs between the source and target instances?**

- If the source instance supports and enables global transaction identifier (GTID), when HA switch occurs in the source instance during incremental sync, the service will be automatically reconnected, and the synced data flow will resume soon after HA switch is completed.
- If HA switch occurs in the target instance during incremental sync, the service will also be automatically reconnected, and the synced data flow will resume soon after HA switch is completed.

## **Can data in an instance on a later version be synced to an instance on an earlier version?**

No. For data sync between instances of the same type, the major version of the target instance must be later than that of the source instance. Taking MySQL as an example, data cannot be synced from MySQL 5.7 to MySQL 5.6.

## **Can the source/target instance be a local instance?**

Yes. Data can be synced from TencentDB to a local database. Supported access types for such sync include public network and CCN.

## **Can I perform two-way sync for DDL statements in a two-way sync topology?**

No. During sync instance creation, DDL statements in only one instance can be synced; otherwise, the algorithm will detect a DDL loop and then prohibit the creation of one of the instances.

## **Are non-transactional engines supported?**

The current technical solution adds the routing information to transactions to mark transaction sources, which relies on the atomicity of transactions. However, such atomicity of databases/tables based on non-transactional engines is corrupted, and the data consistency cannot be guaranteed. Therefore, we recommend you not use non-transactional engines.



# FAQs for Data Subscription Kafka Edition

Last updated : 2021-05-17 18:05:39

## Why can't I consume data?

- Check the network. The address of the Kafka server is a Tencent Cloud private network address, which can only be accessed in Tencent Cloud VPC that is in the same region of the subscribed instance.
- Check whether the subscription topic, private network address, consumer group name, account or password is correct. You can click the subscription name on the [Data Subscription console](#) to go to the subscription details page and consumption management page to view such information.
- Check whether the encryption parameter is correct. For more information, see [What authentication mechanism does Kafka use?](#).

## What is the data format?

Data Subscription Kafka Edition uses Protobuf for data serialization. You can click [here](#) to download the Protobuf file. A demo project also contains the Protobuf file. For more information, see the “Key Demo Logic Description” section of [Data Consumption Demo](#).

## What authentication mechanism does Kafka use?

See the figure below:

```
def block_consume(brokers, topic, group, user, password, trans2sql):
    # create a kafka consumer
    consumer = kafka.KafkaConsumer(topic,
                                    auto_offset_reset='earliest',
                                    enable_auto_commit=False,
                                    group_id=group,
                                    sasl_plain_username=user,
                                    sasl_plain_password=password,
                                    sasl_mechanism='SCRAM-SHA-512',
                                    security_protocol='SASL_PLAINTEXT',
                                    bootstrap_servers=brokers)
```

## When does Kafka commit?

Please first set the `enable_auto_commit` parameter of Kafka as `false` to disable auto commit. The producer inserts a checkpoint message at an appropriate position in the message sequence. After the checkpoint message is consumed, the Kafka client will commit feedback indicating the consumption is completed, so as to ensure message integrity.

## How long are messages in the Kafka client retained? How do I set the consumer offset?

Messages in the Kafka client are retained for 1 day. You can set the `auto_offset_reset` parameter of Kafka as `earliest` or `latest` as needed. If you need to consume data from a specific offset, you can reset the consumer offset with the seek feature of the Kafka client.

# Regular Expressions for Subscription

Last updated : 2020-02-25 15:24:10

## What is a regular expression?

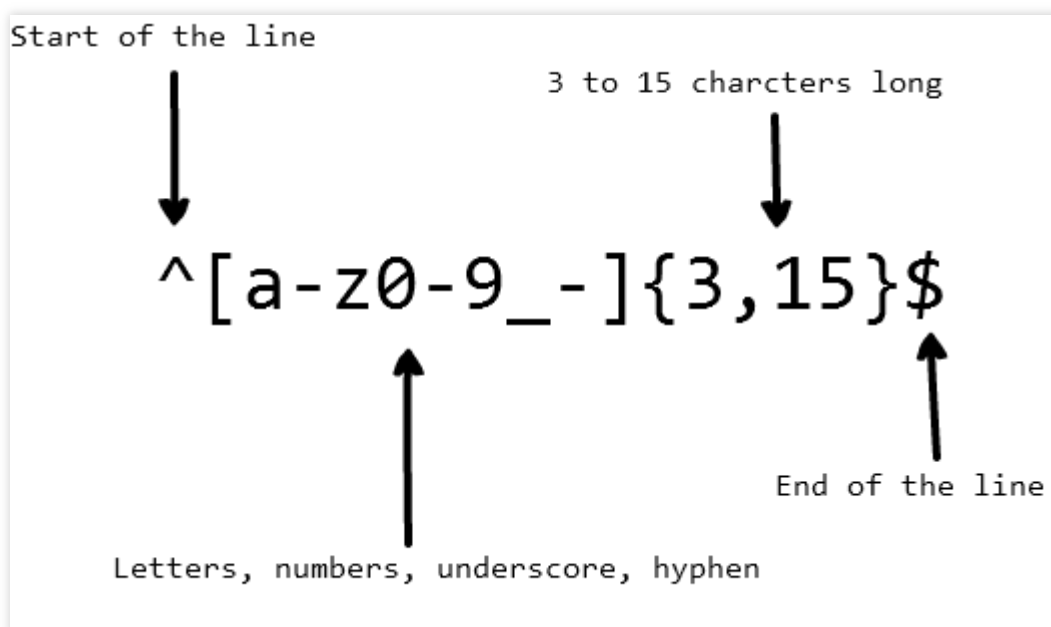
A regular expression is used to search for a specific pattern from text.

A regular expression matches a string from left to right. "Regular expression" is often referred to as "regex" or "regexp" for short.

A regex can be used to replace text in strings, validate forms, extract a substring from a string based on a pattern match, and much more.

If you are developing an application, you may want to set rules on eligible usernames, which can contain letters, digits, underscores, and hyphens.

You may also want to limit the number of characters in a username for better display effect. The following regex can be used to validate a username:



The above regex can match the strings `john_doe` , `jo-hn\_doe` , and `john12\_as` , but not `Jo` as it contains an uppercase letter and is too short.

# Contents

- [Basic Matchers](#)
- [Metacharacters](#)
  - [Period](#)
  - [Character Sets](#)
    - [Negated Character Set](#)
  - [Repetition](#)
    - [Asterisk](#)
    - [Plus Sign](#)
    - [Question Mark](#)
  - [Braces](#)
  - [Capturing Group](#)
  - [Alternation](#)
  - [Escape Character](#)
  - [Anchors](#)
    - [Caret](#)
    - [Dollar Sign](#)
- [Shorthand Character Set](#)
- [Assertion](#)
  - [Positive Lookahead](#)
  - [Negative Lookahead](#)
  - [Positive Lookbehind](#)
  - [Negative Lookbehind](#)
- [Flags](#)
  - [Case Insensitivity](#)
  - [Global Search](#)
  - [Multiline](#)
- [Common Regular Expressions](#)

## Basic Matchers

A regex is just a pattern of characters used to perform a search in text. For example, the regex `cat` means: the letter `c`, followed by the letter `a`, followed by the letter `t`.

```
"cat" => The cat sat on the mat
```

The regex `123` can match the string "123". A regex is matched against the input string by comparing each character in the regex with each character in the input string one by one.

Regexes are normally case-sensitive, so the regex `Cat` would not match the string "cat".

```
"Cat" => The cat sat on the Cat
```

## Metacharacters

Metacharacters are the building blocks of regexes. They do not stand for themselves; instead, they need to be interpreted in certain special ways. Some metacharacters enclosed in square brackets have special meaning. Below are the metacharacters:

Metacharacter	Description
.	Matches any character except a line break.
[ ]	Character class, which matches any character enclosed in square brackets.
[^ ]	Negated character class, which matches any character not enclosed in square brackets.
*	Matches zero or more repetitions of the preceding subexpression.
+	Matches one or more repetitions of the preceding subexpression.
?	Matches zero or one repetition of the preceding subexpression or specifies a non-greedy qualifier.
{n,m}	Braces, which matches the preceding character at least n times but not more than m times.
(xyz)	Capturing group, which matches the character "xyz" in an exact order.
	Alternation, which matches the characters before or after the symbol.
\	Escape character, which can restore the original meaning of metacharacters and allows you to match reserved characters <code>[ ] ( ) { } . * + ? ^ \$ \  </code>
^	Matches the beginning-of-line character.
\$	Matches the end-of-line character.

## Period

The simplest example of metacharacters is the period `.`, which can match any single character but not a line break or newline character. For example, the regex `.ar` means: any character, followed by the letter `a`, followed by the letter `r`.

```
".ar" => The car parked in the garage.
```

## Character Set

A character set is also known as a character class, which is specified by square brackets. A hyphen in a character set is used to specify the character range. The order of the character range inside square brackets does not matter.

For example, the regex `[Tt]he` means: the uppercase letter `T` or the lowercase letter `t`, followed by the letter `h`, followed by the letter `e`.

```
"[Tt]he" => The car parked in the garage.
```

However, the period in character sets is what it means literally. For example, the regex `ar[.]` means: the lowercase letter `a`, followed by the letter `r`, followed by the period `.`

```
"ar[.]" => A garage is a good place to park a car.
```

## Negated Character Set

Generally, the caret symbol `^` represents the start of a string, but when enclosed in square brackets, it negates the character set. For example, the regex `[^c]ar` means: any character except the letter `c`, followed by the character `a`, followed by the letter `r`.

```
"[^c]ar" => The car parked in the garage.
```

## Repetition

The metacharacters `+`, `*`, and `?` are used to specify how many times a subpattern can appear. These metacharacters act differently in different situations.

### Asterisk

The symbol `*` matches zero or more repetitions of the preceding matcher. For example, the regex `a*` matches zero or more repetitions of the preceding lowercase letter `a`. However, if it appears after a character set, then it finds the repetitions of the whole character set.

For example, the regex `[a-z]*` means: any number of lowercase letters in a row.

```
"[a-z]*" => The car parked in the garage #21.
```

The symbol `*` can be used together with the metacharacter `.` to match the arbitrary string `.*`. It can also be used together with the whitespace character `\s` to match a string of whitespace characters.

For example, the regex `\s*cat\s*` means: zero or more whitespaces, followed by the lowercase letter `c`, followed by the lowercase letter `a`, followed by the lowercase character `t`, followed by zero or more whitespaces.

```
"\s*cat\s*" => The fat cat sat on the cat.
```

## Plus Sign

The symbol `+` matches one or more repetitions of the preceding character. For example, the regex `c.+t` means: the lowercase letter `c`, followed by at least one character, followed by the lowercase letter `t`.

```
"c.+t" => The fat cat sat on the mat.
```

## Question Mark

The metacharacter `?` makes the preceding character optional and matches zero or one repetition of the preceding character.

For example, the regex `[T]?he` means: the optional uppercase character `T`, followed by the lowercase letter `h`, followed by the lowercase letter `e`.

```
"[T]he" => The car is parked in the garage.
```

```
"[T]?he" => The car is parked in the garage.
```

## Braces

In regexes, braces, aka quantifiers, are used to specify how many times a character or a group of characters can be repeated. For example, the regex `[0-9]{2,3}` means: match at least 2 digits but not more than 3 digits (characters in the range of 0 to 9).

```
"[0-9]{2,3}" => The number was 9.9997 but we rounded it off to 10.0.
```

The second number can be left out. For example, the regex `[0-9]{2,}` means: match 2 or more digits. If the comma is also removed, the regex `[0-9]{2}` means: match exactly 2 digits.

```
"[0-9]{2,}" => The number was 9.9997 but we rounded it off to 10.0.
```

```
"[0-9]{2}" => The number was 9.9997 but we rounded it off to 10.0.
```

## Capturing Group

A capturing group is a group of subpatterns enclosed in parentheses and is denoted as `(...)`. If a quantifier is placed after a character, it will repeat the preceding character.

However, if a quantifier is placed after a capturing group, it will repeat the whole capturing group.

For example, the regex `(ab)*` matches zero or more repetitions of the string "ab". The metacharacter `|` can be used in a capturing group. For example, the regex `(c|g|p)ar` means: the lowercase letter `c`, `g`, or `p`, followed by the letter `a`, followed by the letter `r`.

```
"(c|g|p)ar" => The car is parked in the garage.
```

## Alternation

The vertical bar `|` is used to define alternation that is like a condition between multiple expressions. Alternation seems to work in the same way as character set.

However, the great difference is that alternation can be used at the expression level, while character set at the character level.

For example, the regex `(T|t)he|car` means: the uppercase character `T` or the lowercase letter `t`, followed by `h`, followed by `e` or `c`, followed by `a`, followed by `r`.

```
"(T|t)he|car" => The car is parked in the garage.
```

## Escape Character

The backslash `\` is used to escape the next character, allowing you to specify a symbol as a matching character including reserved characters `{ } [ ] / \ + * . $ ^ | ?`. To use a special character as a matching character, prepend `\` before it.

For example, the regex `.` is used to match any character except a line break. To match the character `.` in the input string, the regex `(f|c|m)at\.` means: the lowercase letter `f`, `c`, or `m`, followed by the lowercase letter `a`, followed by the lowercase letter `t`, followed by the optional `.` character.

```
"(f|c|m)at\." => The fat cat sat on the mat.
```

## Anchors

Anchors in regexes are used to check whether the matching symbol is the starting or ending symbol of the input string. There are two types of anchors: `^` (which checks whether the matching character is the start character of the input



string) and `$` (which checks whether the matching character is the end character).

### Caret

The caret `^` is used to check whether a matching character is the first character of the input string. If the regex `^a` (if `a` is the starting symbol) is used to match the string `abc`, it matches `a`.

However, if the regex `^b` is used, it does not match anything, because "b" in the string `abc` is not the start character.

The regex `^(T|t)he` means that the uppercase character `T` or the lowercase letter `t` is the starting symbol of the input string, followed by the letter `h`, followed by the lowercase letter `e`.

```
"(T|t)he" => The car is parked in the garage.
```

```
"^(T|t)he" => The car is parked in the garage.
```

### Dollar Sign

The dollar sign `$` is used to check whether a matching character is the last character of the input string. For example, the regex `(at\.)$` means: the lowercase letter `a`, followed by the lowercase letter `t`, followed by the character `.`, and the matcher must be the end of the string.

```
"(at\.)" => The fat cat. sat. on the mat.
```

```
"(at\.)$" => The fat cat sat on the mat.
```

## Shorthand Character Sets

There are shorthands for commonly used character sets and regexes. The shorthand character sets are as follows:

Shorthand	Description
<code>.</code>	Matches any character except a line break
<code>\w</code>	Matches alphanumeric characters: <code>[a-zA-Z0-9_]</code>
<code>\W</code>	Matches non-alphanumeric characters: <code>[^\w]</code>
<code>\d</code>	Matches digits: <code>[0-9]</code>
<code>\D</code>	Matches non-digits: <code>[^\d]</code>

Shorthand	Description
\s	Matches whitespace character: <code>[\t\n\f\r\p{Z}]</code>
\S	Matches non-whitespace character: <code>[^\s]</code>

## Lookaround

Lookbehind and lookahead (also called lookahead) are specific types of **non-capturing groups** (used to match the pattern but not included in the matching list). Lookarounds are used when there is the condition that this pattern is preceded or followed by another certain pattern.

For example, to get all the numbers and the `.` character that are preceded by the character `$` in the input string `$4.44 and $10.88`, the regex `(?<=\$)[0-9\.]*` can be used.

Below are the lookarounds used in regexes:

Symbol	Description
?=	Positive lookahead
?!	Negative lookahead
?<=	Positive lookbehind
?<!	Negative lookbehind

### Positive Lookahead

A positive lookahead asserts that the first part of the expression must be followed by the lookahead expression. The returned match only contains the text that is matched by the first part of the expression.

To define a positive lookahead, parentheses are used. Within those parentheses, a question mark with equal sign is denoted as `(?=...)`. The lookahead expression is written after the equal sign inside parentheses.

For example, the regex `(T|t)he(?=\sfat)` means: the uppercase letter `T` or lowercase letter `t`, followed by the letter `h`, followed by the lowercase letter `e` or `c`.

In parentheses, the positive lookahead is defined, which tells the regex engine to match `The` or `the` which is followed by the word `fat`.

```
"(T|t)he(?=\sfat)" => The fat cat sat on the mat.
```

### Negative Lookahead

A negative lookahead is used to get the content that does not match the expression from the input string and is defined in the same way as positive lookahead.

The only difference lies in that a negative lookahead uses the negation symbol `!` instead of the equal sign `=`, such as `(?!...)`.

For example, the regex `(T|t)he(?!\sfat)` means: get all the words `The` or `the` and add a whitespace character before the unmatched `fat` word from the input string.

```
"(T|t)he(?!\sfat)" => The fat cat sat on the mat.
```

## Positive Lookbehind

A positive lookbehind is used to get all the matches that are preceded by a specific pattern and is denoted as `(?<=...)`. For example, the regex `(?<=(T|t)he\s)(fat|mat)` means: get all the words `fat` and `mat` after the word `The` or `the` from the input string.

```
"(?<=(T|t)he\s)(fat|mat)" => The fat cat sat on the mat.
```

## Negative Lookbehind

A negative lookbehind is used to get all the matches that are not preceded by a specific pattern and is denoted as `(?<!(...))`. For example, the regex `(?<!(T|t)he\s)(cat)` means: get all the `cat` words that are not after the word `The` or `the` from the input string.

```
"(?<!(T|t)he\s)(cat)" => The cat sat on cat.
```

## Flags

Flags are also called modifiers as they modify the output of regexes. They can be used in any order or combination and are an integral part of a regex.

Flag	Description
i	Case-insensitive: Sets matching to be case-insensitive.
g	Global search: Searches for all the matches throughout the input string.
m	Multiline match: Matches every line of the input string.

## Case Insensitivity

The modifier `i` is used to perform a case-insensitive match. For example, the regex `/The/gi` means: the uppercase letter `T`, followed by the lowercase letter `h`, followed by the lowercase letter `e`.

At the end of the regex, the flag `i` tells the regex to ignore the case. As can be seen, the flag `g` is also used so as to search for matches in the whole input string.

```
"The" => The fat cat sat on the mat.  
  
"/The/gi" => The fat cat sat on the mat.
```

## Global Search

The modifier `g` is used to perform a global match (find all matches rather than stopping after the first match).

For example, the regex `/(.at)/g` means: any character except a line break, followed by the lowercase letter `a`, followed by the lowercase letter `t`.

As the flag `g` is used at the end of the regex, it will find all matches in the input string.

```
".(at)" => The fat cat sat on the mat.  
  
"/.(at)/g" => The fat cat sat on the mat.
```

## Multiline

The modifier `m` is used to perform multiline matching. As discussed earlier, anchors (`^`, `$`) are used to check whether the matched character is the beginning or end of the input string. To have anchors work on each line, the flag `m` should be used.

For example, the regex `/at(.*?)$/gm` means: the lowercase letter `a`, followed by the lowercase letter `t`, and optionally zero or one repetition of any character except line break. Because the modifier `m` is at the end of the regex, the regex engine matches pattern at the end of each line in a string.

```
"/.at(.*?)$/gm" => The fat  
                    cat sat  
  
                    on the mat.  
  
"/.at(.*?)$/gm" => The fat  
                    cat sat  
                    on the mat.
```

## Common Regexes

Type	Expression
Positive integer	<code>^\d+\$</code>
Negative integer	<code>^\d+\$</code>
Phone number	<code>^+?[\d\s]{3,}\$</code>
Phone code	<code>^+?[\d\s]+(?[\d\s]{10,})\$</code>
Integer	<code>^-?\d+\$</code>
Username	<code>^\w\d_{4,16}\$</code>
Alphanumeric character	<code>^[a-zA-Z0-9]*\$</code>
Alphanumeric character with whitespace	<code>^[a-zA-Z0-9 ]*\$</code>
Password	<code>^(?=^.{6,}\$)((?=.*[A-Za-z0-9])(?=.*[A-Z])(?=.*[a-z]))^.*\$</code>
Email	<code>^[a-zA-Z0-9._%~+@][a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}\$</code>
IPv4 address	<code>^((?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.){3}(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\$`</code>
Lowercase letters	<code>^[a-z]*\$</code>
Uppercase letter	<code>^[A-Z]*\$</code>
Username	<code>^\w\d_{4,16}\$</code>
URL	<code>^(((http https ftp):\\V)?([[a-zA-Z0-9\\-\\.]+(\\.)?([[a-zA-Z0-9]]{2,4}([[a-zA-Z0-9]V+=%&amp;_\\.~?\\-]*)))*\$</code>
Visa credit card number	<code>^4[0-9]{12}(?:[0-9]{3})?\$</code>
Date (MM/DD/YYYY)	<code>^(0?[1-9] 1[012])[ - /.](0?[1-9] 12)[0-9]{3}[01][ - /.](19 20)?[0-9]{2}\$</code>
Date (YYYY/MM/DD)	<code>^(19 20)?[0-9]{2}[ - /.](0?[1-9] 1[012])[ - /.](0?[1-9] 12)[0-9]{3}[01]\$</code>
Mastercard credit card number	<code>^5[1-5][0-9]{14}\$</code>

# OldDTS

## General

Last updated : 2019-11-20 10:37:37

### **Does DTS support data migration between TencentDB instances under two different Tencent Cloud accounts?**

Yes. For that purpose, you need to log in to the DTS Console with the Tencent Cloud account of the target TencentDB instance and select self-built database with public IP as the source instance type.

### **Can I monitor the progress of database migration tasks?**

Yes. You can log in to the [DTS Console](#) and open the "Data Migration" page to do so.

### **Will DTS delete the data in the source database after migration?**

No. When DTS performs data migration, it simply replicates the data of the source database with no impact on it.

### **Does DTS support scheduled automatic migration?**

Yes. When modifying the configuration for a created data migration task, you can select scheduled migration and specify the start time.

### **Which version of Redis instances can be migrated?**

Source instances on version 3.2 or above cannot be migrated.

### **When will the instance be restarted during MySQL migration?**

- For migration of full instance, the parameters will be synced, and one restart is required for the parameters to take effect.
- For migration of some tables, `replicate_do_table` will be set, and there will be one restart.

### **Will the tables be locked during MySQL migration?**

- For InnoDB, short table locking is required to get the consistent time point after your long-running transactions end.
- For MyISAM, all tables will be locked until cold backup is completed.

### **Which MySQL versions are supported for data migration?**

MySQL 5.1, 5.5, 5.6, and 5.7. As TencentDB no longer supports MySQL 5.1, you are recommended to upgrade MySQL 5.1 to MySQL 5.5 first and then migrate data to TencentDB for MySQL 5.5. You can also use DTS to directly migrate from a local MySQL 5.1 instance to a TencentDB for MySQL 5.5 instance.

In addition, the following restrictions apply:

- Currently, migration is supported between MySQL 5.6 and 5.7 instances but not between MySQL 5.5 and 5.7 instances.
- Virtual columns and JSON in MySQL 5.7 instances are not supported.

### Are TencentDB for MySQL Basic Edition instances supported for migration?

- TencentDB for MySQL Basic Edition instances can be migrated as source instances over the public network but not the private network.
- They cannot be used as target instances currently.

### What if the connectivity check fails?

You can click **View Details** to find a possible solution.

### Why is a target instance unavailable?

Possible reasons include:

1. It is not initialized.
2. It is locked by another task.
3. It has data.
4. Its capacity is smaller than the data volume in the source instance.

### Why does a warning occur when I check a task?

You can click **View Details** on the right of the warning to view the cause and solution.

### Why does an error occur during migration and cause the migration task to fail?

- The BGSAVE operation on the source instance failed during migration.
- During migration, the volume of data written to the source instance is too large and exceeds the configured sync buffer, which causes the migration task to continuously retry establishing the sync connection and generate RDB.

### Why is there a 15-day limit on incremental migration?

Currently, incremental migration is performed through the nearest proxy server via Tencent Cloud Direct Connect, which eliminates network jitters and ensures the quality of data transfer. The 15-day limit can reduce the connection pressure on the proxy server and is only intended for reasonable utilization of resources for migration. Currently, connections will not be force closed after 15 days.

### Why is some data missing in a target instance where the subscription feature is enabled?

As cold backup needs to be imported during migration, `binlog` is disabled for higher write performance, resulting in loss of some data. To prevent data loss, you need to create a migration task first, confirm that the target instance is in sync, and then set the subscription feature for it.

### **What if the TokuDB engine is used in the source instance?**

In this case, TokuDB will be converted to InnoDB by default during migration. As tables containing clustered indices or compressed with TokuDB need to be pre-processed before migration, they are not supported currently. DDL operations on TokuDB are not supported either.

### **Why does the error about super permission occur for full check during migration?**

This is because session needs to be in binlog format for full check, which requires super permission. The solution is as follows:

1. Super permission is not required for spot check.
2. Grant the super permission to the account.

### **Why does an error occur during DTS task check for TokuDB table migration?**

TokuDB cannot be converted to InnoDB if there are tables compressed with TokuDB or containing clustered index in the source instance.



# General

Last updated : 2020-11-19 10:26:52

## **Does the data subscription feature allow multiple SDKs to connect to and consume the same channel simultaneously?**

No. A channel can only be connected to and used by one SDK for consumption at a time. If you have multiple downstream SDKs to subscribe to the same table, please create multiple channels.

## **Does the data subscription feature allow one SDK to connect to multiple channels?**

Yes. An SDK can consume multiple channels at a time.

## **Why does the error "already has sdk on this channel" occur when an SDK is started?**

A channel can only be connected to by one SDK for consumption. If another connection is created, this error will occur. In this case, you need to check whether the application has exited. If the error persists, you can set the restart interval a little longer, such as 20 seconds.

## **Does the real-time incremental data in data subscription refer to the added data or include the modified data?**

The incremental data that can be subscribed to through data subscription includes all INSERT/DELETE/UPDATE changes (DML) and structure changes (DDL).

## **A TencentDB instance and a local database have the same table structure but different indices. Can they be synced in real time by using the data subscription feature?**

Yes. Data subscription only covers data changes, and it does not matter if the indices are different. However, if you have subscribed to structure changes and your TencentDB instance involves index changes, local consumption of structure changes may fail if the indices are different.

## **Why can't I modify the consumption time point of a data subscription channel?**

When this error occurs, relevant prompts will be displayed on the screen. Generally, it is because that the downstream SDK connected to the subscription channel is still consuming data. You can go to the DTS Console to view the consumption source IP and check whether the downstream SDK is consuming data, and if yes, you need to stop the consumption first before you can modify the consumption time point.

## **How can I determine whether data consumption is normal?**

When data is written to a channel (or not all data is consumed), the consumption time point in the console can be migrated normally if data consumption is normal.

## **If a data entry on the consumer side is not acknowledged for data subscription, why does the SDK receive duplicate data entries after restart?**

If there is any message not acknowledged, the SDK will keep pulling it until its cache is full. At this time, the consumption time point saved on the server will be the time point of the last message before it is acknowledged. When the SDK is restarted, the server will push data again from the consumption time point of the unacknowledged message to avoid message loss. Therefore, the SDK will receive some duplicate messages.

## **Why can't data be successfully subscribed to if the SDK is restarted a few days after exit? The error message reads "Maybe checkpoint is too old".**

The data in the data subscription channel will be retained for 1 day and then deleted. If the time point of the last consumed data entry before the SDK exits is -1 day ago, then the data at the consumption time point cannot be subscribed to successfully. To fix this problem, you need to modify the consumption time point to ensure it is within the valid range.

## **When pulling data, the SDK suddenly crashed and couldn't subscribe to any data; after restart, it consumed some data before crashing again. Why?**

This is probably because the `ackAsConsumed` API is not called to report the consumption time point in the SDK code. In this case, the data in the limited cache space of the SDK will not be deleted. When the cache is full, new data cannot be pulled, so the SDK will crash and fail to subscribe to any data.

### **Note :**

All messages here, including BEGIN and COMMIT messages and those irrelevant to the business logic, must be acknowledged for consumption.

## **How do I ensure that the data subscribed to by the SDK is a complete transaction? Will the record in the middle of the transaction be pulled based on the specified consumption time point?**

No. Based on the user-specified consumption time point or the time point of the last acknowledged consumption, the server will search for the start point of the complete transaction corresponding to this consumption time point. Data is sent to the downstream SDK from the beginning of the transaction. Therefore, the full data of the complete transaction can be received.

## **Will any problem in data subscription occur upon TencentDB master/slave switchover or master restart? Will data be lost?**

No. In case of master/slave switchover or master restart, data subscription will automatically perform switch imperceptibly to the SDK.

## Why does the error "Do DTS authentication fail, caused by: get channel info from msg failed" occur when the SDK is started?

You need to verify whether the input parameters are correct, including `ip` , `port` , `secretId` , `secretKey` , and `channelId` .

## Why does the system prompt that `secretId` has no permission when the SDK is started?

A sub-account has no permission by default. It must be granted access to the

`name/dts:AuthenticateSubscribeSDK` operation or all DTS operations through the

`QcloudDTSFullAccess` policy by the root account.

### Note :

You need to create the `QcloudDTSFullAccess` policy on your own as it is not predefined in CAM.

- Grant the SDK access permission to all channels:

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "dts:AuthenticateSubscribeSDK"
      ],
      "resource": "*",
      "effect": "allow"
    }
  ]
}
```

- Grant the SDK access permission to a specified channel:

```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "dts:AuthenticateSubscribeSDK"
      ],
      "resource": "qcs::dts::channel/{channelId}",
      "effect": "allow"
    }
  ]
}
```

**Will duplicate data be received through data subscription?**

No, if data consumption is normal. However, there is a very slim chance that if the SDK quits abnormally and the last acknowledged consumption time point is not reported promptly, duplicate data may be received when the SDK is started next time.

If a complete transaction is not acknowledged, the data will be pulled again from the beginning of the transaction when the SDK is started again. In this case, the data cannot be regarded as duplicate data. The core logic of the SDK will ensure the transaction integrity.

**Can a data subscription instance subscribe to multiple TencentDB instances?**

No. A data subscription instance can only subscribe to one TencentDB instance.

**What should I do if OOM occurs while the SDK is running?**

You are recommended to use a host with higher specification. When a single SDK runs smoothly at a high speed, it consumes less than 1 core of CPU and less than 1.5 GB of memory.