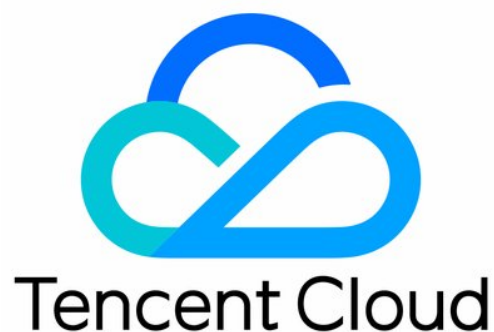


# **Tencent Real-Time Communication Glossary**

## **Product Documentation**



## Copyright Notice

©2013-2019 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.

# Glossary

Last updated : 2020-05-22 12:11:31

## A

### AccountType

`AccountType` is the account type used to log in to a TRTC application in TRTC v1. It is automatically assigned for user authentication after the TRTC application is created and is no longer required in v2.

## F

### Room

A room is an audio/video space where users can receive each other's real-time audio/video streams.

- In TRTC, room is a virtual concept used to separate a group of users from another group.
- A user's audio/video streams can be received only by other users in the same room.
- A user can be in only one room at any time point and can enter another room only after exiting the current room.

#### Note:

- The user who creates a room will be the room owner, but this user cannot close the room actively.
- The server will close the room as soon as all users actively exit the room.
- If a user is disconnected exceptionally, the server will remove this user from the room after 30 seconds. If all users are disconnected exceptionally, the server will automatically close the room after 30 seconds.
- When a user attempts to enter a room that does not exist, the TRTC backend will create the room first and then add the user to the room.

## J

### Role name

After you create a role parameter set for a specified device platform integrated with TRTC v1 in the [TRTC Console](#), you can name this set of parameters, and this name is referred to as a role name. After a user enables the camera or mic and upstreams the audio/video streams in a room, the role to be used during room entry can be specified by the role name, which confirms the parameters for data streaming. This concept is no longer required in v2.

Note:

- During room entry, if a user uses a role name that has not been configured in the console, the user can still enter the room, but the default role will be used.
- After room entry, the user can change the role by calling the role switching API.
- Switching to a non-existent role with the API will throw back a failure callback.
- After the role configuration is modified, logging in to the client again is required to make the new configuration take effect (the SDK will pull the configuration from the backend when login is successful. If the pull fails, the role will not be found).

## P

### Relayed live streaming

Relayed live streaming is a live broadcasting mode that is usually used for data forwarding between audio/video systems with different protocols. In this mode, the TRTC server pushes the audio/video data in a chat room to the LVB service through RTMP push, thereby implementing the features of relayed live streaming, relayed push, On-Cloud MixTranscoding, and On-Cloud Recording.

### PrivateMapKey

`PrivateMapKey` (room ticket) is a credential used for permission verification when a user attempts to enter a room, which is equivalent to the key to a specified room ( `RoomID` ). It is usually used to authenticate room entry and implement room entry restrictions and is issued by your business server. For more information on its generation method, please see [Room Entry Permission Protection](#).

## Q

### Group

The group system in TRTC v1 uses the [group](#) feature of the Tencent Cloud [Instant Messaging \(IM\)](#) service.

Note:

Group and room are two different concepts completely independent of each other.

## R

### RoomID

`RoomID` (room number) is used in TRTC to uniquely identify a room. It is a number in the uint32 range that is maintained and assigned on your own.

## S

### SDKAPPID

`SDKAPPID` (application ID) is a unique ID used by the Tencent Cloud backend to differentiate between different TRTC applications. It is automatically generated when an application is created in the [TRTC Console](#). The data under different `SDKAPPID` is not interconnected.

## Tencent Real-Time Communication

Developed by Tencent Cloud based on Tencent's over a decade of expertise in QQ audio/video chat technologies, Tencent Real-Time Communication (TRTC) provides cross-platform, high-quality, and customizable real-time communication services by leveraging the WebRTC capabilities of Tencent Browsing Service (TBS) and TRTC SDKs. It enables you to build an audio/video communication platform from scratch quickly through Mobile QQ, WeChat Mini Program, WeChat Official Account, QQ Browser, and over 20,000 TBS-enabled apps with no knowledge of audio/video technologies required.

## T

### TRTC

For more information, please see [TRTC](#).

## U

## UserID

`UserID` (user ID) is used to uniquely identify a user in a TRTC application.

- `UserID` maps a user account in your business system to Tencent Cloud. Usually, you can directly use a username as the `UserID`.
- The recommended length of `UserID` is below 32 bytes. It can be a combination of letters, digits, and underscores but cannot only contain digits. It is case insensitive.

## UserSig

`UserSig` (user signature) is used for user login authentication to make sure that the user's identity is authentic. For more information on its generation method, please see [Signature Generation](#).