# Brekeke PBX

## Version 2

## Log Database Developer’s Guide

Brekeke Software, Inc.

Version

Brekeke PBX Version 2 Log Database Developer's Guide

Revised May 2011

Copyright

This document is copyrighted by Brekeke Software, Inc.

Copyright © 2011 Brekeke Software, Inc.

This document may not be copied, reproduced, reprinted, translated, rewritten or readdressed in whole or part without expressed, written consent from Brekeke Software, Inc.

Disclaimer

Brekeke Software, Inc. reserves the right to change any information found in this document without any written notice to the user.

Trademark Acknowledgement

- *LINUX is a registered trademark of Linus Torvalds in the United States and other countries.*
- *Red Hat is a registered trademark of Red Hat Software, Inc.*
- *Windows is a trademark or registered trademark of Microsoft Corporation in the United States and other countries.*
- *Mac is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.*
- *Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.*
- *Other logos and product and service names contained in this document are the property of their respective owners.*

# 1. Introduction

This document explains how to record Brekeke PBX's log information using a database. In this document, MySQL database is used as the example database.

# 2. Table

## 2.1. t_pbx_ua_log

Log information is generated for every call between Brekeke PBX and UAs. There will be two records per call.

```
    CREATE TABLE t_pbx_ua_log (
        c_tenant_id                 INT,
        c_id                        BIGINT,
        c_seq                       SMALLINT,
        c_direction                 TINYINT,
        c_type                      TINYINT,
        c_pbxuser                   TINYINT,
        c_result                    TINYINT,
        c_rescode                   SMALLINT,  /* since v2.4 */
        c_q850code                  SMALLINT,  /* since v2.4 */
        c_ars                       VARCHAR(50),
        c_phoneid                   VARCHAR(50),  /*since v2.4.9.0*/
        c_number                    VARCHAR(50),
        c_pbx_number                VARCHAR(50),
        c_user                      VARCHAR(50),
        c_start                     DATETIME default NULL,
        c_start_talking             DATETIME default NULL,
        c_end                       DATETIME default NULL
```

The below indexes will be needed. Based upon your requirements, add additional indexes for other columns as needed.

```
CREATE UNIQUE INDEX idx_pbx_ua_log_id on t_pbx_ua_log (c_tenant_id,c_id,c_seq);
```

### 2.1.1. c_tenant_id

This ID represents a tenant. This ID is used to identify tenants when Brekeke PBX Multi-Tenant Edition is used to host multiple tenants under one server. When multiple Brekeke PBX servers are used to feed in information into one database, set a unique ID for each tenant at the PBX servers.

### 2.1.2. c_id

This ID represents a call. This ID is generated for every call. In environments where multiple tenants exist under the same server, there can be duplicate *c_id values* for different tenants. Each call can be differentiated by looking at c_tenant_id values associated with the *c_id*.

### 2.1.3. c_seq

This ID is generated when a new row is added to the database. These IDs may not always be in sequence; sometimes ID values may be skipped from the order.

### 2.1.4. c_direction

“1” represents an incoming session to Brekeke PBX. “2” represents an outgoing session from Brekeke PBX.

### 2.1.5. c_type

This ID represents the type of call handled by Brekeke PBX.

| | | |
|---|---|---|
| 0 | missed | missed call (Call was not established) |
| 1 | in | call received by Brekeke PBX |
| 2 | out | call initiated by Brekeke PBX |
| 3 | transfer | transferred call (transferred using Auto Attendant or Dial Pad command) |
| 4 | call-pickup | call pickup feature is used |
| 5 | park-pickup | call picked up after call park |
| 6 | hold-pickup | call picked up after call hold |
| 7 | join | joined conversation |
| 8 | supervisor | supervise feature is used |
| 9 | resume | resume a call which had been disconnected while a call is on hold |

### 2.1.6. c_pbx_user

“1” represents a Brekeke PBX user. “0” represents an external user.

### 2.1.7. c_result

Result codes.

| | |
|---|---|
| 0 | in progress |
| 1 | connected (include when the call is establishing) |
| 2 | busy |
| 3 | cancel |
| 4 | not found (ex. phone number can not be located) |
| 5 | timeout (can not be connected) |

### 2.1.8. c_rescode

This ID represents the SIP response code generated when a call has been rejected. This column will be used only when cti.logging.type3.rescode=true is set at Brekeke PBX Admintool > [Options]>[Advanced]. (Available on version 2.4 or later)

### 2.1.9. c_q850code

This ID represents the Q.850 code generated when a call has been rejected. This column will be used only when cti.logging.type3.q850code=true is set at Brekeke PBX Admintool > [Options]>[Advanced]. (Available on version 2.4 or later)

### 2.1.10. c_ars

ARS route name that was used. When an ARS route is not used, “” (zero-length string) will be entered.

### 2.1.11. c_phoneid

The phone number used by each UA in the call. (Available for version 2.4.9.0 or later)

### 2.1.12. c_number

UA’s number.

### 2.1.13. c_pbx_number

The other party’s number that Brekeke PBX shows to the UA.

#### 2.1.14. c_user

User associated with the call (User who will be billed for the call)

#### 2.1.15. c_start

Time when a session started.

#### 2.1.16. c_start_talking

Time when conversation started (responded to call). NULL is stored while ringing or if the call was never established.

#### 2.1.17. c_end

Time when a session ended. During a call, NULL is stored.

### 2.2. t_pbx_call_log

Log recorded by each call. This table will be used when parameters for each call needs to be saved. Currently this table is available only for calls with PSTN lines.

```
CREATE TABLE t_pbx_call_log (
     c_tenant_id                INT,
     c_id                       BIGINT,
     c_main_seq                 SMALLINT,
     c_sparam1                  VARCHAR(255),
     c_iparam1                  INT,
     c_iparam2                  INT
  ) DEFAULT CHARSET=utf8;
```

The below indexes will be needed. Upon your requirements, add additional indexes for other columns.

```
CREATE UNIQUE INDEX idx_pbx_call_log_id on t_pbx_call_log (c_tenant_id,c_id);
```

#### 2.2.1. c_tenant_id

This ID represents a tenant. This ID is used to identify tenants when Brekeke PBX Multi-Tenant Edition is used to host multiple tenants under one server. When multiple Brekeke PBX servers are used to feed in information into one database, set a unique ID for each tenant at the PBX servers.

### 2.2.2. c_id

This ID represents a call. This ID is generated for every call. In environments where multiple tenants exists under the same server, there can be duplicate *c_id values* for different tenants. Each call can be differentiated by looking at c_tenant_id values associated with the *c_id*

### 2.2.3. c_main_seq

*c_seq* in *t_pbx_ua_log* for external session of the call

### 2.2.4. c_sparam1

Used for applications

### 2.2.5. c_iparam1

Used for applications

### 2.2.6. c_iparam2

Used for applications

# 3. Settings

1. Place JDBC driver (jar file) under the directory: <pbx-installdir>/webapps/pbx/WEB-INF/lib
2. Set below settings from Brekeke PBX Admintool > [Options]>[Advanced].
3. Restart the Brekeke PBX to activate the settings.

## 3.1. Setting Example

```
com.brekeke.pbx.pbxplugin=com.brekeke.cti.logging.type3.LoggingPluginType3, com.brekeke.pbxco
nsole.StatusBoard
cti.logging.type3.missedlog.all=true
cti.logging.type3.user=<database user>
cti.logging.type3.password=<database password>
cti.logging.type3.driver=com.mysql.jdbc.Driver
cti.logging.type3.url=jdbc:mysql://<database-hostname>/<database name>?useUnicode=true&chara
cterEncoding=UTF8&autoReconnect=true
```

## 3.2. Property

### 3.2.1. com.brekeke.pbx.pbxplugin

*com.brekeke.cti.logging.type3.LoggingPluginType3,com.brekeke.pbxconsole.StatusBoard*

This property is necessary to have log exported using the method described in this document.

### 3.2.2. cti.logging.type3.calllog

When set to “false”, log will not be exported to *t_pba_call_log*. Default is set for “true”.

### 3.2.3. cti.logging.type3.driver

JDBC driver name for connecting database.

### 3.2.4. cti.logging.type3.missedlog.all

Set whether log for incomplete calls will be exported or not. Default is set for “true”.

### 3.2.5. cti.logging.type3.missedlog.ars

Set a route name using regular expression. Incomplete calls for the specified route will be included in the log export. This property should be used when *cti.logging.type3.missedlog.all* is set for “false”. This setting overrides settings at *cti.logging.type3.missedlog.use.in* and *cti.logging.type3.missedlog.user.out*.

### 3.2.6. cti.logging.type3.missedlog.user.in

Set one or a group of user names using regular expression. Incomplete calls initialized from the particular user(s) will be included in the call log export. This property should be used when *cti.logging.type3.missedlog.all* is set for “false”.

### 3.2.7. cti.logging.type3.missedlog.user.out

Set one or a group of user names using regular expression. Incomplete calls sent to the particular user(s) will be included in the call log export. This property should be used when *cti.logging.type3.missedlog.all* is set for “false”.

### 3.2.8. cti.logging.type3.password

Password for connecting database.

### 3.2.9. cti.logging.type3.phoneid

Set whether the phone number used by UAs will be exported or not. Default is set for “false”.

### 3.2.10. cti.logging.type3. pstn.length

The minimum string length recognized as external PSTN phone numbers. This parameter is used for setting *c_main_seq* at *t_pbx_call_log*. Default value is “10”.

### 3.2.11. cti.logging.type3.q850code

When set as “true*”, c_q850code* column in t_pbx_ua_log will be used. Default is set for “false”.

### 3.2.12. cti.logging.type3. rescode

When set as “true”, *c_rescode* column in *t_pbx_ua_log* will be used. Default is set for “false”

### 3.2.13. cti.logging.type3.ualog

When set to “false”, log will not be exported to *t_pbx_ua_log* table. Default is set for “true”.

### 3.2.14. cti.logging.type3.url

URL for connecting database.

### 3.2.15. cti.logging.type3.user

User name for connecting database.