

Brekeke PBX

Version 3

Log Database Developer's Guide

Brekeke Software, Inc.

Version

Brekeke PBX version 3 Log Database Developer's Guide

Copyright

This document is copyrighted by Brekeke Software, Inc.

Copyright © 2012 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 United States and other countries.*
- ◆ *Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.*
- ◆ *Other logos and product and service names contained in this document are the properties of their respective owners.*

1.	INTRODUCTION	4
2.	TABLE	4
2.1.	T_PBX_UA_LOG	4
2.1.1.	<i>c_tenant_id (Multi-tenant PBX)</i>	5
2.1.2.	<i>c_id</i>	5
2.1.3.	<i>c_seq</i>	5
2.1.4.	<i>c_direction</i>	5
2.1.5.	<i>c_type</i>	5
2.1.6.	<i>c_pbx_user</i>	6
2.1.7.	<i>c_result</i>	6
2.1.8.	<i>c_rescode</i>	6
2.1.9.	<i>c_q850code</i>	6
2.1.10.	<i>c_ars</i>	7
2.1.11.	<i>c_phoneid</i>	7
2.1.12.	<i>c_number</i>	7
2.1.13.	<i>c_real_number</i>	7
2.1.14.	<i>c_pbx_number</i>	7
2.1.15.	<i>c_user</i>	7
2.1.16.	<i>c_real_pbx_number</i>	7
2.1.17.	<i>c_start</i>	7
2.1.18.	<i>c_start_talking</i>	8
2.1.19.	<i>c_end</i>	8
2.2.	T_PBX_CALL_LOG	8
2.2.1.	<i>c_tenant_id (Multi-tenant PBX)</i>	8
2.2.2.	<i>c_id</i>	9
2.2.3.	<i>c_main_seq</i>	9
2.2.4.	<i>c_sparam1</i>	9
2.2.5.	<i>c_iparam1 and c_iparam2</i>	9
3.	SETTINGS.....	9
3.1.	SETTING EXAMPLE	9
3.2.	PROPERTY	10
3.2.1.	<i>com.brekeke.pbx.pbxplugin</i>	10
3.2.2.	<i>cti.logging.type3.calllog</i>	10
3.2.3.	<i>cti.logging.type3.driver</i>	10

- 3.2.4. *cti.logging.type3.missedlog.all* 10
- 3.2.5. *cti.logging.type3.missedlog.ars* 10
- 3.2.6. *cti.logging.type3.missedlog.user.in* 10
- 3.2.7. *cti.logging.type3.missedlog.user.out* 10
- 3.2.8. *cti.logging.type3.password* 10
- 3.2.9. *cti.logging.type3.pbxid* 11
- 3.2.10. *cti.logging.type3.phoneid* 11
- 3.2.11. *cti.logging.type3.pstn.length* 11
- 3.2.12. *cti.logging.type3.q850code* 11
- 3.2.13. *cti.logging.type3.real_number* 11
- 3.2.14. *cti.logging.type3.rescode* 11
- 3.2.15. *cti.logging.type3.ualog* 11
- 3.2.16. *cti.logging.type3.url* 11
- 3.2.17. *cti.logging.type3.user* 11

1. Introduction

This document explains how to record Brekeke PBX's log information using a database. In this document, MySQL 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 are two records per call.

```
CREATE TABLE t_pbx_ua_log (  
    c_tenant_id          VARCHAR(20), /* Multi-tenant PBX */  
    c_id                 BIGINT,  
    c_seq               SMALLINT,  
    c_direction         TINYINT,  
    c_type              TINYINT,  
    c_pbxuser           TINYINT,  
    c_result            TINYINT,  
    c_rescode           SMALLINT,  
    c_q850code          SMALLINT,  
    c_ars               VARCHAR(50),  
    c_phoneid           VARCHAR(50),  
    c_number            VARCHAR(50),  
    c_real_number       VARCHAR(50),  
    c_pbx_number        VARCHAR(50),  
    c_user              VARCHAR(50),  
    c_real_pbx_number   VARCHAR(50),  
    c_start             DATETIME default NULL,  
    c_start_talking     DATETIME default NULL,  
    c_end               DATETIME default NULL  
    ) DEFAULT CHARSET=utf8;
```

The indexes below will be required. Depending on your requirements, you may add more 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 (Multi-tenant PBX)**

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

Starting with Brekeke PBX version 3, there is no [Tenant ID] field in each Tenant > [Options] setting page from Brekeke PBX admintool. By default, tenant name will be put in column *c_tenant_id* when Brekeke PBX log database is in use.

To display [Tenant ID] field and insert value of [Tenant ID] in Brekeke PBX log database, add the following parameter in file **config.properties** and restart the Brekeke PBX server machine:
config.properties is at <Brekeke PBX Installation Directory>/webapp/pbx/WEB-INF/work/pbx/

com.brekeke.pbx.Setting.TENANT_ID=true

2.1.2. **c_id**

This ID represents a call. It 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 the *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, because ID values in the sequence may sometimes be skipped.

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	call is resumed after being disconnected while call was 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 (includes when the call is establishing)
2	busy
3	cancel
4	not found (e.g., phone number cannot be located)
5	timeout (call cannot 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 with the following parameter set at Brekeke PBX Admintool > [Options] > [Advanced]

```
cti.logging.type3.rescode=true
```

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 with the following parameter set at Brekeke PBX Admintool > [Options] > [Advanced]

```
cti.logging.type3.q850code=true
```

2.1.10. c_ars

ARS route name that is applied to the call.

When an ARS route is not used, "" (zero-length string) will be entered.

2.1.11. c_phoneid

For an incoming session to Brekeke PBX, this field is the phone number set in Brekeke PBX for the caller. If no phone ID can be applied to the call, it will be the real number used by UA.

For an outgoing session from Brekeke PBX, this field is the phone number set in Brekeke PBX for the called PBX users. If no phone ID can be applied to the call, it will be the real number dialed by UA.

2.1.12. c_number

The user number set in Brekeke PBX for this UA.

2.1.13. c_real_number

For an incoming session to Brekeke PBX, this field is the real number used by UA.

For an outgoing session from Brekeke PBX, this field is the real number to which Brekeke PBX sent call.

2.1.14. c_pbx_number

For an incoming session to Brekeke PBX, this field is the user number to which Brekeke PBX sent the call. This number may not be the same as the number dialed by the caller.

For an outgoing session from Brekeke PBX, this field is the caller's PBX user number.

2.1.15. c_user

User associated with the call (i.e., user who will be billed for the call).

2.1.16. c_real_pbx_number

For an incoming session to Brekeke PBX, this field is the real number dialed by the caller.

For an outgoing session from Brekeke PBX, this field is the caller number shown to the recipient.

This number may not be the same as the caller's PBX user number.

2.1.17. c_start

Time when a session starts.

2.1.18. c_start_talking

Time when the conversation starts (i.e., when the recipient responds to the call).

NULL is stored while UA is ringing or if the call is never established.

2.1.19. c_end

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

2.2. t_pbx_call_log

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

```
CREATE TABLE t_pbx_call_log (
    c_tenant_id          VARCHAR(20), /* Multi-tenant PBX */
    c_id                 BIGINT,
    c_main_seq          SMALLINT,
    c_sparam1           VARCHAR(255),
    c_iparam1           INT,
    c_iparam2           INT
) DEFAULT CHARSET=utf8;
```

The indexes below will be required. Depending on your requirements, you may add more indexes for other columns as needed.

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

2.2.1. c_tenant_id (Multi-tenant PBX)

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

Starting with Brekeke PBX version 3, there is no [Tenant ID] field in each Tenant > [Options] setting page from Brekeke PBX admintool. By default, tenant name will be put in column *c_tenant_id* when Brekeke PBX log database is in use.

Check the setting in table *t_pbx_ua_log* about how to display [Tenant ID] field.

2.2.2. `c_id`

This ID represents a call. It 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 the `c_tenant_id` value 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` and `c_iparam2`

Used for applications.

3. Settings

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

3.1. Setting Example

```
com.brekeke.pbx.pbxplugin=com.brekeke.cti.logging.type3.LoggingPlugin
Type3, com.brekeke.pbxconsole.StatusBoard
cti.logging.type3.missedlog.all=true
cti.logging.type3.phoneid=true
cti.logging.type3.real_number=true
cti.logging.type3.pbxid=false
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&characterEncoding=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 export a log 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 to true.

3.2.3. **cti.logging.type3.driver**

JDBC driver name for connecting database.

3.2.4. **cti.logging.type3.missedlog.all**

Set whether the log for incomplete calls will be exported. Default is set to true.

3.2.5. **cti.logging.type3.missedlog.ars**

Set a route name using a 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 to false.

This setting overrides the settings at:

cti.logging.type3.missedlog.user.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 a 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 to false.

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

Set one or a group of user names using a 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 to false.

3.2.8. **cti.logging.type3.password**

Password for connecting database.

3.2.9. cti.logging.type3.pbxid

Set whether the real session id of the call will be exported. Default is set to false.

3.2.10. cti.logging.type3.phoneid

Set whether the phone number used by UAs will be exported. Default is set to false.

3.2.11. 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 set to 10.

3.2.12. cti.logging.type3.q850code

When set to true, *c_q850code* column in *t_pbx_ua_log* will be used. Default is set to false.

3.2.13. cti.logging.type3.real_number

Set whether real numbers in incoming or outgoing sessions will be exported to table *t_pbx_ua_log* column *c_real_number* and *c_real_pbx_number*. Default is set to false.

3.2.14. cti.logging.type3.rescode

When set to true, *c_rescode* column in *t_pbx_ua_log* will be used. Default is set to false.

3.2.15. cti.logging.type3.ualog

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

3.2.16. cti.logging.type3.url

URL for connecting database.

3.2.17. cti.logging.type3.user

User name for connecting database.