

OnDO PBX

Version 1.4

**Call Log Plug-in Developer's Guide
(For Integration with Billing System)**

Brekeke Software, Inc.

Version

OnDO PBX v.1.4 Call Log Plug-in Developer's Guide, July 2005

Copyright

This document is copyrighted by Brekeke Software, Inc.

Copyright ©2005 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.....	3
2.	SYSTEM REQUIREMENT	3
3.	CREATING A CALL LOG PLUG-IN.....	3
3.1.	Arguments in the Method	3
4.	SAMPLE PROGRAM.....	5
5.	INSTALLATION.....	6

1. Introduction

Since v1.2, OnDO PBX has had a Call Log function to record call history. The Call Log function can work with other applications by creating a plug-in. You will need to write Java programs to create plug-ins. This document provides Java developers a guide on how to create a Call Log plug in for the OnDO PBX product.

2. System Requirement

The system requires you to use JDK1.4 or later.

3. Creating a Call Log Plug-in

Steps

- 1) For building your code, add:
`<OnDO PBX install directory>/WEB-INF/lib/ondopbx.jar` in the classpath.
- 2) Implement `com.brekeke.pbx.plugin.CallLogWriter` interface in your class.
- 3) `CallLogWriter` interface

There is only one method in the interface.

```
public void logging (long starttime, long endtime, String[] data);
```

3.1. Arguments in the Method

- `starttime`
Start time of the call (Start time of ringing)
- `endtime`
End time of the call

- data

Data for each session. This includes Call Log data that is almost the same as what you see in the [CallLog] menu of OnDO PBX Admin tool. Each element of this array is a data of one session and contains the following data with comma-delimited format:

- PBX user If this call is a PBX user's call, the user's name will be written here.
- Route name When an ARS rule is applied, the route name will be written here.

Call type (numeric value)

- | | | |
|----|--------------|---|
| 0. | missed* | The session was not established |
| 1. | caller | The session is a caller side (who initiated the call) session |
| 2. | callee | The session is a callee side (who received the call) session |
| 3. | transfer | The session is for the person who transferred the call |
| 4. | call-pickup | The session is for the person who did Call Pick Up |
| 5. | park-pickup | The session is for the person who picked up a parked call |
| 6. | hold-pickup* | The session is for the person who picked up a hold call |
| 7. | join* | The sessions is for the person who joined a call |
| 8. | supervisor* | The session is for the person who supervised a call |
| 9. | resume* | The session was resumed after hanging up a call on hold |

✓ **These call types are only available for the version later than v1.4.*

- The other party's SIP URI
- This party's SIP URI
- Session Start time (INVITE) (long value)
- Talk start time (long value)
- Talk time (ms)

✓ *The characters #, , ; % \$ \ are escape characters. (Add \ before each escape character).*

An instance of the plug-in class will be created when your PBX starts. The method of the instance "logging" will be called every time a call ends.

4. Sample Program

The following section shows a sample program that writes call data into a specified file.

```
package yourpackage;
public class LogPrintWriter implements CallLogWriter {
    private PrintWriter w;
    public LogPrintWriter(){
        try{
            w = new PrintWriter( new FileOutputStream( "C:/log.txt", true ) );

        }catch( IOException e ){
            // error
        }
    }

    public synchronized void logging(long starttime, long endtime, String[] data) {
        try{
            w.print( starttime );
            w.print( "-" );
            w.println( endtime );
            for( int i = 0; i < data.length; i++){
                w.println( data[i] );
            }
            w.flush();
        }catch( Exception e ){
            // error
        }
    }

    public void finalize() throws Throwable {
        try{
            super.finalize();
            w.close();
        }catch( Exception e ){
        }
    }
}
```

5. Installation

Steps

- 1) Put your new plug-in class in the Classpath

Place the compiled class under:

<OnDO PBX Install dir>/WEB_INF/classes/your_package_directories

OR

Compress the class file into a JAR file and place it under <OnDO PBX install dir>/WEB-INF/lib

- 2) Add a property for the plug-in

Open <OnDO PBX install dir>/WEB-INF/work/pbx/setting.properties with a text editor. Add the following property for the new plug-in's class (with the package name) in the property file and save it. `com.brekeke.pbx.Setting.LOGWRITER`

Example)

```
com.brekeke.pbx.Setting.LOGWRITER= yourpackage.LogPrintWriter
```

- 3) Restart the machine on which OnDO PBX is installed. Your OnDO PBX will start working with your new plug-in after the restart.