

Brekeke PBX

Version 3

Audio File Plug-in Developer's Guide

Brekeke Software, Inc.

Version

Brekeke PBX Version 3 Audio File Plug-in Developer' s Guide

Copyright

This document is copyrighted by Brekeke Software, Inc.

Copyright © 2013 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.*
- ◆ *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. INTRODUCTION4
- 2. STEPS TO CREATE A PLUG-IN.....4
- 3. INTERFACE4
 - 3.1. INTERFACE: RECORDINGPLUGIN, RECORDINGPLUGIN_MT.....4
 - 3.1.1. Method: eventRecorded..... 4
- 4. SAMPLE.....5
- 5. INSTALLATION.....6
 - 5.1. PUT YOUR CLASS INTO A CLASSPATH.....6
 - 5.2. SET YOUR CLASS NAME IN THE PROPERTY FILE6
 - 5.2.1. Use your class for all PBX users 6
 - 5.2.2. Use your class for a specific user 6
- 6. DEFAULT PLUG-INS7
 - 6.1. CLASS: RECORDINGFILEHTTPUPLOADER7
 - 6.1.1. Settings 7
 - 6.2. CLASS: RECORDINGFILEMOVE.....8
 - 6.2.1. Settings 8

1. Introduction

This plug-in interface provides ways to process the audio files created by the call recording or voicemail features of Brekeke PBX. You need to use Java programming language to create your own plug-in. The following plug-ins are included in the product. You don't need to develop plug-in when uploading to http server or moving files.

2. Steps to Create a Plug-in

- 1) Add the ondopbx.jar file to your classpath:
`<Brekeke PBX install_dir>/webapps/pbx/WEB-INF/lib/ondopbx.jar`
- 2) Create a JAVA class and implement the interface:
`com.brekeke.pbx.media.plugin.RecordingPlugin.`

3. Interface

3.1. Interface: RecordingPlugin, RecordingPlugin_mt

Package name: com.brekeke.pbx.media.plugin

With Brekeke Multi-Tenant PBX, "RecordingPlugin_mt" will be used.

3.1.1. Method: eventRecorded

Description:

This method is called when recording finishes and an audio file is created. An instance of the class will be created every time a message file is created.

Parameters:

| | | |
|--------|---|----------------------|
| tenant | Tenant name (available only for Interface:RecordingPlugin_mt) | |
| user | User | |
| uprop | Properties related to the user | |
| file | File created | |
| prop | Properties of the file | |
| type | Type | |
| | TYPE_VOICEMAIL(0) | Voicemail |
| | TYPE_CONVERSATION_RECORDING(1) | Recorded call |
| | TYPE_NAME (2) | Name |
| | TYPE_GREETING1 (3) | Personal Greeting |
| | TYPE_GREETING2 (4) | Alternative Greeting |

Returns:

To delete the file, return false. To save the file, return true.

4. Sample

The following is a sample program which will upload the recorded calls to a FTP server. Nothing will be performed when voicemails are created.

(Please refer to other resources regarding the detailed codes for FTP)

```
package yourpackage;

import java.io.*;
import java.util.*;

import com.brekeke.pbx.media.plugin.RecordingPlugin_mt;

public class YourRecordingPlugin implements RecordingPlugin {

    public boolean eventRecorded( String tenant, String user, Properties userProp,
        File file, Properties prop, int type) {

        if( type == TYPE_CONVERSATION_RECORDING ){
            SampleFTPUploader.uploadToFtpServer( file );
            return false;
        }else{

            return true;
        }
    }
}
```

5. Installation

5.1. Put your class into a classpath

Place your compiled class file in exact directory structure of package name under the directory:

<Brekeke PBX install_directory> /webapps/pbx/WEB-INF/classes

For example:

class name = YourClass, package name = com.yourdomain

Place “YourClass.class” under the directory:

<Brekeke PBX install_directory> /webapps/pbx/WEB-INF/classes/com/yourdomain

or compress your class file into a jar file and place the jar file into

<Brekeke PBX install directory>/webapps/pbx/WEB-INF/lib

5.2. Set your class name in the property file

5.2.1. Use your class for all PBX users

From Brekeke PBX Admintool > [Options] > [Advanced], add the following property

com.brekeke.pbx.Setting.RECORDING_PLUGIN

After restarting the Brekeke PBX, this change will take effect.

Example:

com.brekeke.pbx.Setting.RECORDING_PLUGIN = yourpackage.YourRecordingPlugin

5.2.2. Use your class for a specific user

Open the specific user.properties file with text editor. Set the property “media.recording.plugin” with your class name including the package name. Save the file. In this case, restarting Brekeke PBX is not necessary.

Each user’s user.properties file is at

<Brekeke PBX install directory>/webapps/pbx/WEB-INF/work/pbx/users/<userID>/

Example:

media.recording.plugin = yourpackage.YourRecordingPlugin

6. Default Plug-ins

6.1. Class: RecordingFileHttpUploader

Package name: com.brekeke.pbx.plugin.recording

After every recording, Brekeke PBX send recorded file with POST requests to HTTP server. The file is sent in multipart/form-data format. Recorded data is in u-Law format.

Properties:

The following properties will be sent with POST requests when a recording file is uploaded.

| | |
|--------|---|
| tenant | Tenant name |
| rid | Call' s ID |
| tid | ID that identify SIP UA and a specific call |
| to | User that records calls |
| from | Number of the Callee |
| fileid | file name (no extension displayed) |

6.1.1. Settings

Add the following parameters (Java Properties format) at Brekeke PBX Admintool > [Options] > [Advanced] page.

com.brekeke.pbx.Setting.RECORDING_PLUGIN

Set "com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader",
such as:

```
com.brekeke.pbx.Setting.RECORDING_PLUGIN=  
com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader
```

com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader.URL

Specify URL of HTTP server where recording file will be uploaded.

com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader.PARAMS

When adding parameters, set parameters in the format "a=b&c=d"

com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader.VOICEMAIL

Set "true" here when you are uploading recorded voicemail message files (default value=true)

com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader.CALL_RECORDING

Set “true” here when you are uploading recorded files using call recording feature (default value=true)

com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader.OTHER_TYPE

Set “true” here when you are uploading recorded files created for personalized greetings or name (default value=false)

com.brekeke.pbx.plugin.recording.RecordingFileHttpUploader.KEEP_COPY

Set “true” here when you want to leave the recording files after uploading to the HTTP server (default value=false)

6.2. Class: RecordingFileMove

Package name: com.brekeke.pbx.plugin.recording

After every recording, Brekeke PBX move recorded files to a specified location. Recorded data is in u-Law format. When moving the files to a different machine, you’d need to assign the folder at the different machine as a network drive of the server Brekeke PBX is installed on. (With Linux OS, you’d need to mount an NFS file system. The recorded files will be moved one by one, Brekeke PBX will not move multiple files at once.

6.2.1. Settings

Add the following parameters (java property format) at Brekeke PBX Admintool > [Options] > [Advanced] page.

com.brekeke.pbx.Setting.RECORDING_PLUGIN

Set “com.brekeke.pbx.plugin.recording.RecordingFileMove”

com.brekeke.pbx.plugin.recording.RecordingFileMove.DEST_PATH

Specify file path where recorded files are moved to. Replace path name that is enclosed by “#”.

Properties that can be replaced:

| | |
|--------|---|
| tenant | Tenant name |
| rid | Call's ID |
| tid | ID that identify SIP UA and a specific call |
| to | User that records calls |
| from | Number of the Callee |

fileid file name (no extension displayed)
index When index is set at parameters, Brekeke PBX will update file names
 by adding number from 1 and add up.

Ex. c:/root/#tenant#/#to#/#rid#_#index#.ul

With this example setting, by setting “#index#”, the files recorded in one call have unique name. As a result, Brekeke PBX will move each recorded files to specified location.

com.brekeke.pbx.plugin.recording. RecordingFileMove.VOICEMAIL

Set “true” here when you are moving saved voice message files. (default value=true)

com.brekeke.pbx.plugin.recording. RecordingFileMove.CALL_RECORDING

Set “true” here when you are moving recorded files using call recording feature (default value=true)

com.brekeke.pbx.plugin.recording. RecordingFileMove.OTHER_TYPE

Set “true” here when you are moving recorded files created for personalized greetings or name (default value=false)

com.brekeke.pbx.plugin.recording. RecordingFileMove.MAX_INDEX

The maximum value for index (default value=10)