



VCLogWebSDK

Installation & Configuration

Guide

Version 02
Released 2014-09-23

All Right Reserved © Voicocyber Technologies Ltd. 2014.

With non-written permission from Voicocyber Technologies Ltd, any units and individuals are not allowed to excerpt, copy the contents of this document some or all of, and cannot be transfer in any form.

Trademark Notice



VoiceCyber and other Voicocyber Technologies trademarks are the Voicocyber Technologies Ltd trademarks.

All other trademarks or registered trademarks mentioned in this document are owned by respective holders.

Notice

The contents of this document are subject to change without notice. Every effort has been made to ensure the accuracy of this document. However, due to ongoing product improvement and revisions, VoiceCyber cannot guarantee the accuracy of the material after the date of publication nor can it accept responsibility for errors or omission. VoiceCyber will publish updates and revisions to this document as needed.

No part of this material may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from VoiceCyber Technologies Ltd.

VoiceCyber Technologies Ltd

Website: <http://www.voicectber.com>

Content

1 SDK Minimum Requirement	1
Minimum Requirement for SDK.....	1
2 Installing SDK	2
3 Field Mapping	1
3.1 Update Customer Data.....	1
3.1.1 Method to Update Customer Data.....	1
3.1.2 Parameter Description.....	2
3.1.3 Configuration File Description	2
3.2 Search Recording Information	3
3.2.1 Methods	3
3.2.2 Parameter Description.....	3
3.2.3 Configuration File Description	3
3.3 Get Recording Playback Address by Reserved Field	4
3.3.1 Method	4
3.3.2 Parameter Description.....	4
3.4 Update Recording Data Table.....	5
3.4.1 Introduction.....	5
3.4.2 Instruction.....	5
3.5 Send CallInfo to Voice.....	5
3.5.1 Introduction.....	5
3.5.2 Instruction.....	5
3.6 Configure VCLogWebSDK.exe.config.....	6

1 SDK Minimum Requirement

Minimum Requirement for SDK

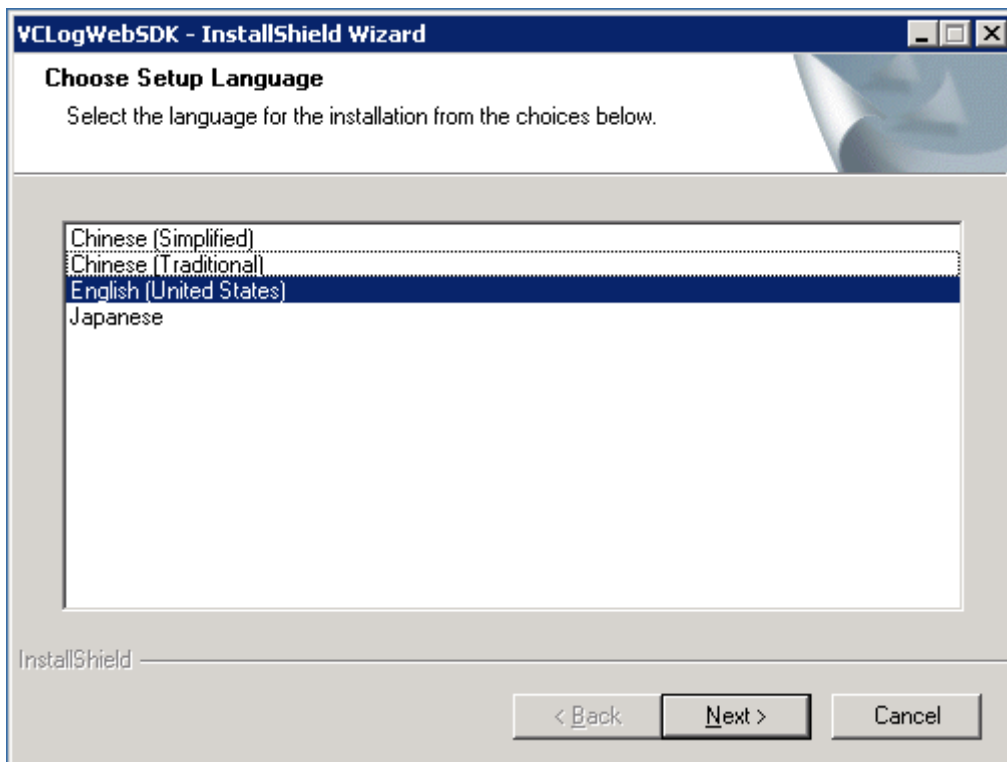
- At least net framework 3.5 sp1 is installed.

- If database-using Mysql, MySQL ODBC 5.1 Driver is required; If using Oracle database, at least Oracle server or client 8.1.7 is required.

2 Installing SDK

Step 1 Double Click on “VCLogWebSDK.exe” and run SDK installation wizard. Following windows appear as shown in **Error! Reference source not found.** to select languages. Click “Next” after select the language.

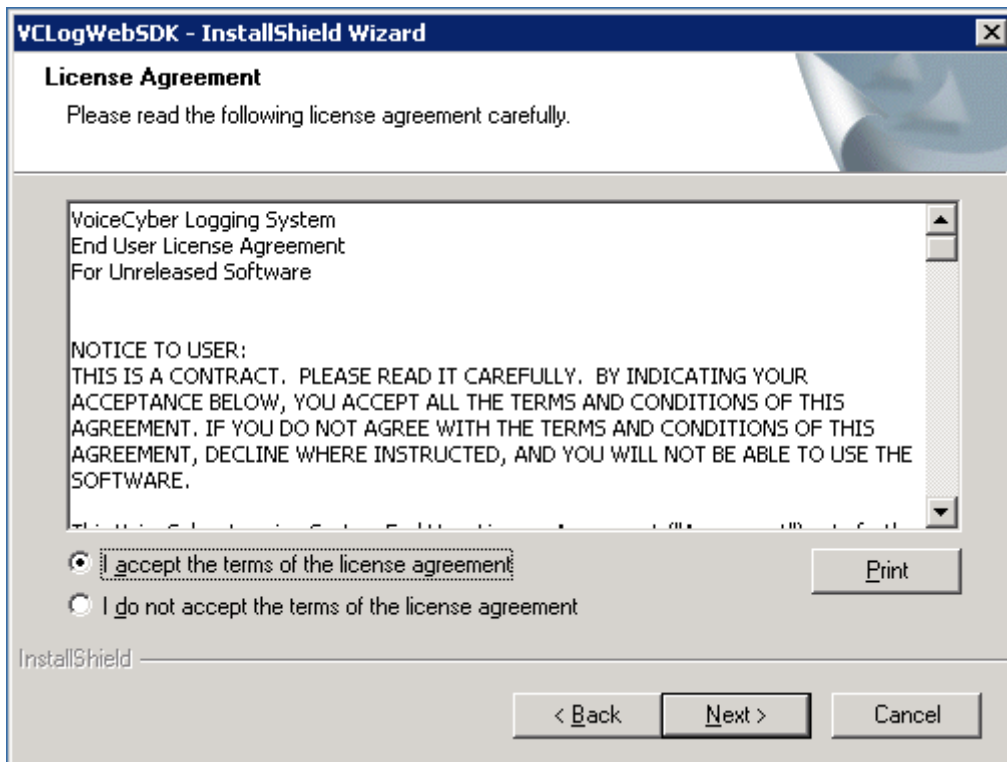
Figure 2-1 Choose Setup Language



Step 2 Welcome screen appear, click “Next”.

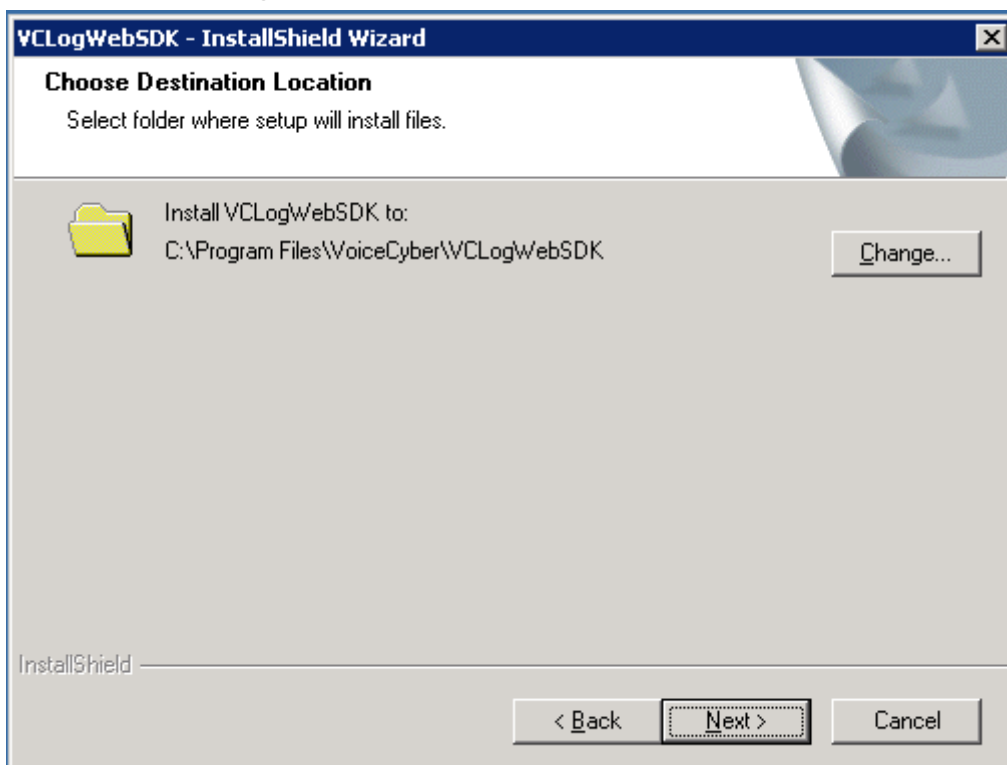
Step 3 License Agreement windows appears as shown in **Error! Reference source not found.** check “I accept the terms of license agreement”, click “Next”.

Figure 2-2 License Agreement



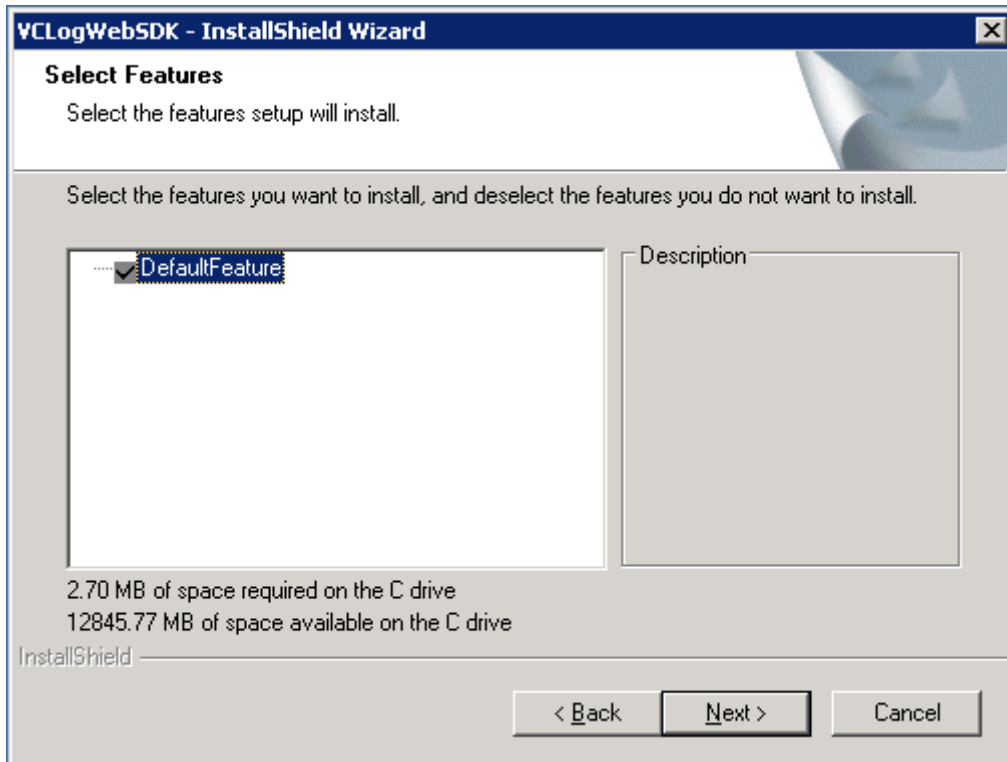
Step 4 Choose Destination Location windows appear, select your desired installation location and then click "Next". User also can select default installation destination, click "Next" as shown in **Error! Reference source not found..**

Figure 2-3 Choose Installation Destination



Step 5 Select Features windows appear, currently only have SDK components, click “Next” without making any changes.

Figure 2-4 Select Features



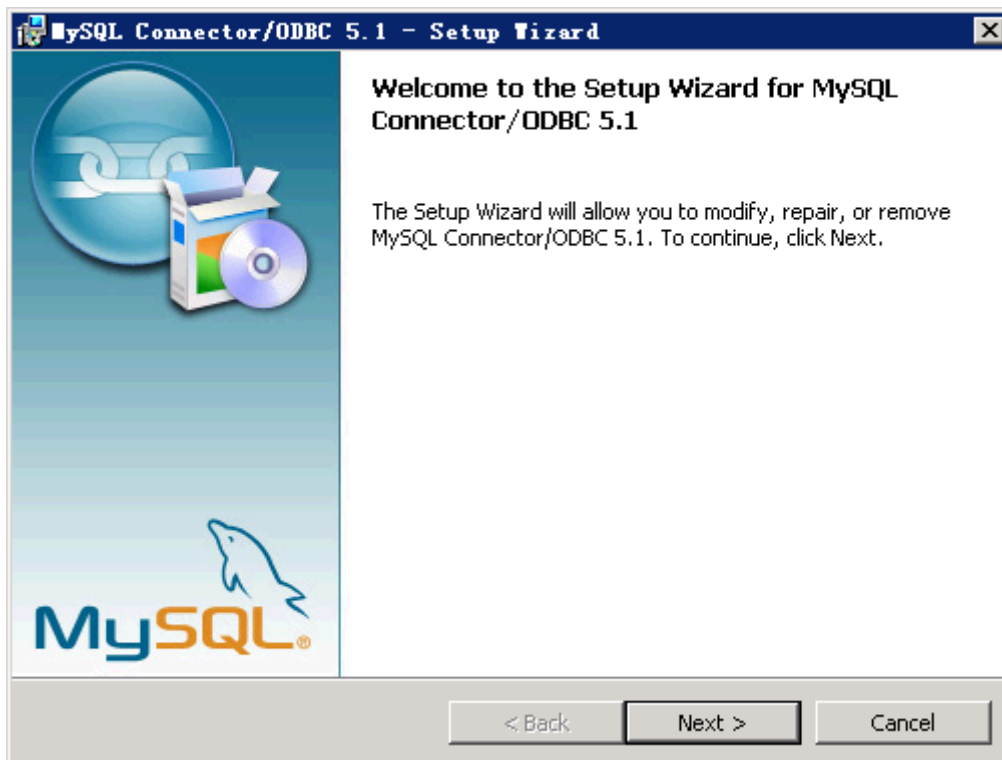
CAUTION

If .NET framework 3.5.1 is not install, “Please install .NET Framework 3.5.1” warning will appear when click “Next”. User can select to install .NET Framework 3.5.1 to proceed, or cancel the installation, to manually install the .NET Framework 3.5.1, then restart the VCLogWebSDK setup again.

Step 6 Installation wizard windows appear, click “Install”.

Step 7 MySQL ODBC5.1 installation windows appear, as shown in Figure 2-5. Click “Next”.

Figure 2-5 MySQL ODBC5.1



Step 8 License Agreement window appear, check accept the terms, click “Next”.



CAUTION

If MySQL ODBC5.1 is already installed, it will directly navigate to select installation type page, user can click “Repair”, or cancel MysqlODBC5.1 installation.

Step 9 Select Installation type window appear, select “Typical”, and click “Next”.

Step 10 “Ready To Install” windows appear, click “Install”.

Step 11 Configuration window appears as show in Figure 2-6

Figure 2-6 SDK Service Configuration

Table 2-1 SDK Service Configuration

	Parameter	Description
Service Configuration	SDK Service Address	3 rd party can refer to this URL to use WebSDK WCF service. During configuration, change the 127.0.0.1 to the SDK service actual IP address.
	Language	Configure configuration windows language, English, Simplified Chinese, Traditional Chinese and Japanese.
Database Configuration	Using the Same Configuration	VMC database column, recording database column, database type in CPe database column, database IP, database port number, username, and password will use the SDK database parameters,. Check this options, the 3 databases will used the same configuration as SDK database, and these 3 column is not editable except for database name.
	SDK Database	SDK database, need to create database for first time installation

	VMC Database	VMC database for recording server, SDK can obtain recording parameter from this database.
	Recording Database	Vclogdb database for recording server, SDK required this database when search recording.
	CPCe Database	CPCe Databse, SDK required this database when updating customer data. If CPCe is not implemented, will fill in based on VCLogdb database.
Other Configuration	Media IP	VCLogMedia service IP, please fill in actual IP address.
	Media Port	VCLogMedia service port, default is 2122.
	License Service Address	License service IP address.
	License Service Port	License service port number, default is 3070.
	Monitoring Recording Server	Recording server monitor by SDK, Click "Get Voice IP" after configure VMC parameter, will obtain voice recording server list, and then select the server to be monitored. As shown in Figure 2-7
	Update Original Data	If checked, it will enable update original data feature. Uncheck to disable this option. Update database table please refer to chapter 一) 1.2。
	CF Table Name	CPCe Customer Data table name, default as record_cf
	CF Reference Name	Reference column name of customer data table in CPCe database, default as recordreference。
	Send Call Info	If checked, if client using Update Original data method, WebSDK will send updated information to VCLogVoice. After VCLogVoice received information, itwill be send to VClog DBBridge when call end. DBBridge will write the updated information to RecordOriginalData table.



CAUTION

- 1, "Update Original Data" and "Send CallInfo to Voice" has the same purpose, but way of execution is different. Enable both method at the same time will waste system resources and reduce WebSDK performance.
 - 2, Using "Send CallInfo to Voice", updated information transfer must be after recording start and before recording end. It is because by using websdk-voice-dbbriidge method, voice only take care the custom data which is after recording startand before recording end.
 - 3, "Update Original Data" function is to mapped and update the data of record_cf based on originaldatafieldmapping.xml format into recordoriginaldata table.
 - 4, In general, whenever client successfully calls update method, WebSDK will write the custom information into record_cf table.
-

Step 13 According to Table 2-1, after configure SDK configuration. Click "Create" in SDK Database column, "Database Created" pop-up message appear when database is created.

Step 14 Click "Test Connection" on each SDK Databse, VMC Database, Recording Database and CPCE Database, to check on database connection. It will prompt "Database is connected". If failed, please check the database configuration.



CAUTION

Without CPCE in environment, fill in VCLogDB parameters including database name in "CPCE Database" configuration. Users need to manually add record_cf table, currently only supported MySQL and MSSQL. SQL script to create record_cf can be found at CF_SQL folder in WebSDK installation directory.

Step 15 Click "Get VoiceIP", will obtain the recording server list. Check the recording server related with SDK and click "Apply", "Successfully Modified and Restart Service" message appears. "Services restart successfully" when services are restarted, and service state will turns green.

Step 16 Close the configuration windows, done for VCLogSDK installation and configuration.

----End

Figure 2-7 Recording Server Monitoring

The screenshot displays the 'SDK Service Configuration' window, which is organized into three main sections: Service Configuration, Database Configuration, and Other Configuration.

Service Configuration: This section includes the 'SDK Service Address' field, which is populated with 'http://127.0.0.1:9000/VLogWebSDK/WebSDK', and a 'Language' dropdown menu currently set to 'U.S. English'.

Database Configuration: This section is divided into four columns, each representing a different database: SDK Database, VMC Database, Recording Database, and CPCE Database. A checkbox labeled 'Using the Same Configuration' is present at the top left of this section. Each column contains fields for 'Database Type' (a dropdown menu), 'Server', 'Port', 'Database Name', 'User', and 'Password' (masked with dots). Below these fields are 'Test Connection' and 'Create' buttons.

Other Configuration: This section contains various settings including 'Media IP' (127.0.0.1), 'Media Port' (2122), 'License Service Address' (127.0.0.1), 'License Service Port' (3070), 'CTI Port' (3020), 'Update Original Data' (unchecked), 'CF Table Name' (record_cf), 'CF Reference Name' (RecordReference), 'Monitoring Recording Server' (with a 'Get Voice IP' button), and 'Send CallInfo to Voice' (unchecked). A list box at the bottom of this section shows '192.168.1.161' with a checked selection icon.

Service State: Located at the bottom left, it features a green circular indicator and the text 'Service State'.

Buttons: At the bottom right, there are four buttons: 'Import', 'Export', 'Apply', and 'Close'.

3

Field Mapping



CAUTION

In environment without CPCe, please follow part 3.3 Get Playback Address based on Reserved Field to configure ReservedFieldMapping.xml, configuration for minor part is not required.

In VCLogWebSDK installation path, there are total 4 configuration files. VCLogWebSDK.exe.config : WebSDK service configuration file, please don't manually modify this file. Please use VCLogWebSDKConfig.exe to perform configuration. Modification of this file required restarts of WebSDK service to take effect. Other 4 configuration files responsible to update customer data, search recording data, get recording file address by Reserved Field and Synchronize record_cf table field mapping. Field mapping can be configured through FieldMapConfig node.

FieldName : Field name, actual name used in database.

FieldType : Field type, field type used in database, such as varchar.

CustomerName : Field name defined by customer.

CustomerType : Field type defined by customer such as varchar or numeric.

FieldMapConfig: Multiple nodes can be configured based on requirement.

3.1 Update Customer Data

3.1.1 Method to Update Customer Data

1. Update Data based on Extension: WebSDKReturn UpdateRecordDataExt(string Extension, string Data);
2. Update Data based on Reference ID: WebSDKReturn UpdateRecordDataByID(string RefID, string Data);

3. StartRecord: WebSDKReturn StartRecord(string Extension, string Data);
4. StopRecord: WebSDKReturn StopRecord(string Extension, string Data);
5. PauseRecord: WebSDKReturn PauseRecord(string Extension, string Data);
6. ResumeRecord: WebSDKReturn ResumeRecord(string Extension, string Data);

3.1.2 Parameter Description

Incoming data as Json format string, update multiple field required to separate by comma, e.g:
 {"CustomerID":"10012345","AgentID":"53201" }

3.1.3 Configuration File Description

1. Customer data mapping is configured in CustomerFieldMapping.xml under VLogWebSDK installation directory. File content included CF1 as one node, FieldName as field name in Recor_cf table, FieldType as field type in Recor_cf table. CustomerName as customer 3rd party system field name, used to call VLogWebSDK methods for updating the data using json format string.

2. Format as shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<FieldListConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<Fields>
    <FieldMapConfig FieldName="CF1" FieldType="varchar" CustomerName="C1"
CustomerType="varchar" />
    <FieldMapConfig FieldName="CF2" FieldType="varchar" CustomerName="C2"
CustomerType="varchar" />
</Fields>
</FieldListConfig>
```

3. Add node : If adding mapping is required, user can add the FieldMapConfig sub-node within Fields nodes in CustomerFieldMapping.xml.



CAUTION

VCLogWebSDK field mapping basic principle is: first to match with mapping in XML, if no content in XML, then will match with databases field. As shown in No.2 format, customer can use C1 and C2, but try not to use CF1 and CF2, and CF3 is exist in record_CF, due to no CF3 mapping, all the customer can directly use CF3.

3.2 Search Recording Information

3.2.1 Methods

WebSDKReturn GetRecordList(string from, string end, string JsonOtherConditions);

3.2.2 Parameter Description

Incoming JsonOtherConditions, combined caller & called ID and other search condition to form Json string, e.g: {"Ext": "1001", "Caller": "5301"}

3.2.3 Configuration File Description

1. Search Condition data mapping is configured in SelectConditionMapping.xml under VCLogWebSDK installation directory. By default, file content included Agent as a node. FieldName as field name in RecordOriginalData table, FieldType as field type in RecordOriginalData table. CustomerName as customer 3rd party system field name, used to call VCLogWebSDK methods for updating the data using json format string.

2. Format as shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<FieldListConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Fields>
    <FieldMapConfig FieldName="AgentID" FieldType="varchar" CustomerName="Agent"
CustomerType="varchar" />
    <FieldMapConfig FieldName="Recordlength" FieldType="varchar" CustomerName="length"
CustomerType="varchar" />
  </Fields>
</FieldListConfig>
```

3. Add Node : If adding mapping is required, user can add the FieldMapConfig sub-node

within Fields nodes in SelectConditionMapping.xml.

3.3 Get Recording Playback Address by Reserved Field

3.3.1 Method

WebSDKReturn GetFilePathbyReservedFields(string reservedFields);

3.3.2 Parameter Description

Incoming reservedFields maybe is the standard string, and maybe is json standard string, such as: {"ReservedThree":"0x2313"}. When incoming data is standard string, it will be mapped into one of the FieldMapConfig node field in ReservedFieldMapping.xml. If first FieldMapConfig node is not exist or ReservedFieldMapping.xml file is not available, then default will be ReservedFour

1. ReservedField data mapping is configured in ReservedFieldMapping.xml under VCLogWebSDK installation directory. By default, file content included ReservedFour as a node. FieldName as reserved field name, FieldType as reserved field type. CustomerName as customer 3rd party system field name, used to call VCLogWebSDK methods for updating the data using json format string.

1. Format as shown below:

```
<?xml version="1.0" encoding="utf-8"?>

<FieldListConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Fields>

    <FieldMapConfig FieldName="ReservedFour" FieldType="varchar" CustomerName="RF4"
CustomerType="varchar" />

    <FieldMapConfig FieldName="ReservedThree" FieldType="varchar" CustomerName="RF3"
CustomerType="varchar" />

  </Fields>

</FieldListConfig>
```

2. Add Node: If adding mapping is required, user can add the FieldMapConfig sub-node within Fields nodes in ReservedFieldMapping.xml.

3.4 Update Recording Data Table

3.4.1 Introduction

All SDK update customer data method will insert all the customer data into record_cf table, every 1 minutes interval, SDK will based on the field mapping configured in OriginalDataFieldMapping.xml, synchronize the data from record_cf into recordoriginaldata table.

3.4.2 Instruction

1. There is OriginalDataFieldMapping.xml under SDK installation directory, Default file content synchronized CF1 to ReservedFour. FileName as field in recordoriginaldata table, FileType as field type, CustomerName as one of the field in record_cf.

2. Format as shown below:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<FieldListConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <Fields>
```

```
    <FieldMapConfig FieldName="ReservedFour" FileType="varchar" CustomerName="CF1"
CustomerType="varchar" />
```

```
    <FieldMapConfig FieldName="ReservedThree" FileType="varchar" CustomerName="CF4"
CustomerType="varchar" />
```

```
  </Fields>
```

```
</FieldListConfig>
```

3. Add Node : If adding mapping is required, user can add the FieldMapConfig sub-node within Fields nodes in OriginalDataFieldMapping.xml.

3.5 Send CallInfo to Voice

3.5.1 Introduction

Enable “Send CallInfo to Voice”, when client call update method, WebSDK will get the custom message from client. The message is convert based on CallInfoMapping.xml into Voice recognizable CTI message and send to VCLogVoice service. VCLogVoice will send the message to VCLogDBBridge service before recording end, and DBBridge will write the data into recordoriginaldata.

3.5.2 Instruction

1. CallinfoMapping.xml located at SDK installation directory. By default, CFI is synchronized to reserverFour. CustomerName is refer to message field name send from the client. FileType and

CustomerType is refer to field type, FieldName is refer to field name that receive the call information.

2. Format as shown below:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<FieldListConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <Fields>
```

```
    <FieldMapConfig FieldName="ReservedFour" FiledType="varchar" CustomerName="CF1"
CustomerType="varchar" />
```

```
    <FieldMapConfig FieldName="ReservedThree" FiledType="varchar" CustomerName="CF4"
CustomerType="varchar" />
```

```
  </Fields>
```

```
</FieldListConfig>
```

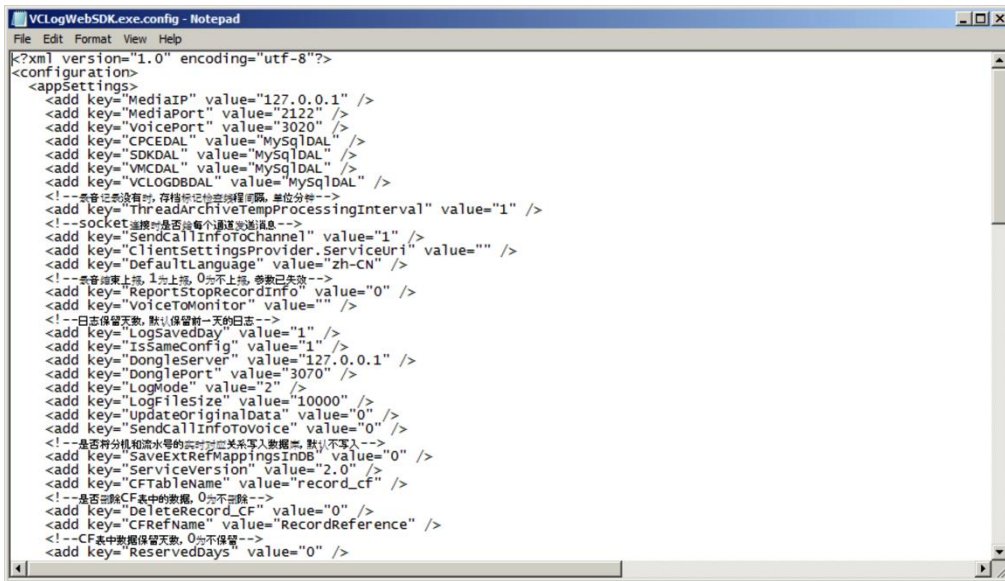
3. Add Node : If adding mapping is required, user can add the FieldMapConfig sub-node within Fields nodes in OriginalDataFieldMapping.xml.

---- End Field Mapping Configuration

3.6 Configure VCLogWebSDK.exe.config

Step 1 Open VCLogWebSDK.exe.config in VCLogWebSdk instllation directory, as shown in Figure 3-1

Figure 3-1 VCLogWebSDK.exe.config



Step 2 Configure paramters, description as shown Table 3-1

Table 3-1 Description

Parameter	Description
ThreadArchiveTempProcessingInterval	Configure interval to flag archive thread if no recording, unit:minutes
SendCallInfoToChannel	Send message to all channel when socket is connected.
ReportStopRecordInfo	Report when recording stop. 1 to report, 0 to disable.
LogSavedDay	Configure number of days to keep log. Log will be deleted after preset day.
SaveExtRefMappingsInDB	Mapped Extension and ReferenceID and write into DB.
ServiceVersion	WebSDK version, default as 2.0
ReservedDays	Number of day to keep data in CF table
SaveUpdateDatainXMLFile	When request Update method, save CF data in XML file first.
VoiceVersion	Refer to recording version, can be configured to version 6 or 7.

Parameter	Description
TryUpdateRecordTimes	Configure retry limit after update original data table failed, default is 30
UpdateOriginalDataTime	Update Original Data interval, default as 60 seconds.

Step 3 Save Configuration Files.

----End