

Voice Client

Version 9.0.21305

Workforce Connect



ZEBRA

Programmer Guide

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. ©2021 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: zebra.com/linkoslegal.

COPYRIGHTS: zebra.com/copyright.

WARRANTY: zebra.com/warranty.

END USER LICENSE AGREEMENT: zebra.com/eula.

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Contents

About this Guide	4
Chapter Descriptions.....	4
Document Conventions.....	4
Notational Conventions.....	5
Service Information.....	5
Revision History.....	5
Actions	7
Load New Configuration File.....	7
Update Configuration Parameters.....	8
Update with Token.....	8
WFC Voice Sign-out Action.....	9
Quit.....	10
Make a Call.....	11
End a Call.....	13
Import Contacts.....	13
Phone State Monitoring.....	15
Headless Mode	17
Enabling Headless Mode.....	17
WFC Voice Connector	18
WFCDemo.....	18

About this Guide

This guide provides application programming interface (API) information for writing applications that use WFC Voice. The guide assumes that the reader is familiar with WFC Voice intents.

The API consists of a series of intents that can be invoked via the following methods:

1. **StageNow:** A WFC Voice intent can be generated using **StageNow** as described at techdocs.zebra.com/mx/intent/. The intent can be sent immediately, scheduled, or triggered off by pressing a button or sensor.
2. **MDM:** A WFC Voice intent can be generated from many MDMs. A couple of examples are provided below:
 - **Soti Mobicontrol:** www.soti.net/mc/help/v14.4/en/scriptcmds/reference/sendintent.html
 - **Airwatch/Workspace ONE :** docs.vmware.com/en/VMware-Workspace-ONE-UEM/2008/ProdProv_All/GUID-FBB62176-0AE9-40CD-89CD-553E9BE04D95.html
3. **WFC Voice application:** A WFC Voice application can generate an intent to another application on the device. For more information, refer to developer.android.com/guide/components/intents-filters.

When WFC Voice receives the intent, it takes the action indicated by the intent. The available intents are detailed in the following sections and include initiating Voice calls, sending messages, signing in/out of the application, and configuring the client.

Chapter Descriptions

The following chapters cover in this guide:

- [About this Guide](#) chapter provides explanation about the document conventions, notational conventions, service information, related documentation, and revision history.
- [Actions](#) chapter provides information about enabling background voice configuration.
- [Headless Mode](#) provides information about configuring the PTT Pro application.
- [WFC Voice Connector](#) provides information about WFCvoiceConnector library and initiating and controlling voice call remotely.

Document Conventions

- **Extras:** Many of the intents have extras, a standard part of Android intents. These are indicated in the following tables by “Extra 0”, “Extra 1”, etc., followed by the type, name, and acceptable values.

- **URI Parameters:** Several of the intents include a URI which may include query parameters. These are indicated in the following tables by `URI parameter 0`, `URI parameter 1`, etc., followed by the name and acceptable values.

Notational Conventions

The following conventions are used in this document:

- **Bold** text is used to highlight the following:
 - Dialog box, window, and screen names
 - Drop-down list and list box names
 - Checkbox and radio button names
 - Icons on a screen
 - Key names on a keypad
 - Button names on a screen
- Bullets (•) indicate:
 - Action items
 - List of alternatives
 - Lists of required steps that are not necessarily sequential.
- Sequential lists (for example, those that describe step-by-step procedures) appear as numbered lists.

Service Information

If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: zebra.com/support.

When contacting support, please have the following information available:

- Serial number of the unit
- Model number or product name
- Software type and version number

Zebra responds to calls by email, telephone, or fax within the time limits set forth in support agreements.

If your problem cannot be solved by Zebra Customer Support, you may need to return your equipment for servicing and will be given specific directions. Zebra is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty.

If you purchased your Zebra business product from a Zebra business partner, contact that business partner for support.

Revision History

Revision	Date	Description
MN001718A01, Rev A	02/2016	Initial release.

About this Guide

Revision	Date	Description
MN001718A02, Rev A	05/2019	Added new section, Signing Out Registered Users.
MN001718A03, Rev A	06/2021	Updated for WFC Voice version 9.0.21109.
MN001718A04, Rev A	11/2021	Updated the Make Call intents, versions supported and UI changes.

Actions

This section provides descriptions and methods to initiate actions available in the WFC Voice API.

Load New Configuration File

The WFC Voice app reads the XML configuration file and restarts services with the new configuration. This is used during initial deployment or after an upgrade.

Prerequisites

- WFC Voice Client is loaded on the device.
- An XML file with configuration parameters has been prepared and is available.
- All versions of WFC Voice Client 9.0 support these intents.



NOTE: The URI may be a network location or a file location on the device.

Intent Definition

Name	Description
Action	wfc.voice.ACTION_NEW_CONFIG
Intent Type	startActivity
Extra 0	This extra parameter defines the URI of the configuration file.
Type	String
Name	profile_uri
Value	URI pointing to file location

ADB Examples

File located on the network:

```
$ adb shell am start -a wfc.voice.ACTION_NEW_CONFIG --es profile_uri https://example.com/voice/profiles/1234
```

File located on the device:

```
$ adb shell am start -a wfc.voice.ACTION_NEW_CONFIG --es profile_uri /
sdcard/new_profile.xml
```

Update Configuration Parameters

The WFC Voice app updates parameters and reloads services. It may be simpler to use this intent when updating one or two parameters, as an option to loading a new configuration file.

Prerequisites

- WFC Voice Client is loaded on the device.
- All versions of WFC Voice Client 9.0 support these intents.

Intent Definition

Name	Description
Action	wfc.voice.ACTION_UPDATE_CONFIG
Intent Type	startActivity
Extra 0	This extra parameter specifies the configuration parameter and value. This extra parameter may be present once or multiple times to update multiple configuration parameters.
Type	String
Name	Parameter(s) as defined in the appropriate WFC Voice PBX Administrator Guide.
Value	Corresponding value



NOTE: The URI may be a network location or a file location on the device.

ADB Example

Updating the log level in the client:

```
$ adb shell am start -a wfc.voice.ACTION_UPDATE_CONFIG --es log_level Error
```

Update with Token

This intent is used to pass a token to the WFC Provisioning Manager for the following purposes:

- Receive configuration information associated with the token in return. After application is successfully configured, the application restarts. For more information on specific configuration parameters related to PBX, refer to the WFC Voice PBX Administrator Guide.
- Associate WFC Voice Client with a customer account for licensing the client. The token may or may not have configuration information associated with it.

Prerequisites

- The WFC Provisioning Manager must be set up with the token used in the intent. Refer to the [Zebra WFC Provisioning Manager Customer Administrator Guide](#) for more information on tokens and associated configuration information with them.
- All versions of WFC Voice Client 9.0 support these intents.

Intent Definition

Name	Description
Action	<code>android.intent.action.VIEW</code> with URI and package
Intent Type	<code>startActivity</code>
URI	<code>wfcvp://<token></code>
Package	<code>com.symbol.wfc.voice</code>



NOTE: Upon a fresh installation of the application, WFC Voice contacts the WFC Provisioning Manager, using its device ID as the token, and receives any configuration information associated with its device ID in return.

ADB Example

```
$ adb shell am start -a android.intent.action.VIEW -d "wfcvp://ACME-3016"
com.symbol.wfc.voice
```

WFC Voice Sign-out Action

The `wfc.voice.SIGN_OUT` action achieves different goals depending on the presence and value of the change parameter. It can be used to invoke the following behavior in the WFC Voice Client:

- **Reload** (formerly called Sign out/Reload): Use with Extension Manager to sign out and reload the configuration based on the configuration parameters or use it to simply force WFC Voice Client to re-register when configured without Extension Manager.
- **Change Department:** Use with Extension Manager to sign out and reload the configuration with a list of all available departments. Users can select new departments to register with multiple extensions. In legacy configurations, this is the same as reload.
- **Add Department:** Use with Extension Manager to sign out and reload the configuration with a list of all available departments, where the originally configured department is auto-selected. Users can add or change departments to register with multiple extensions. In legacy configurations, this is the same as reload.
- **Sign Out:** Use to sign out and de-register from the PBX. The app remains signed-out until the user manually restarts the app or the app receives an external intent to reload. Sign Out functions the same with or without Extension Manager.

Prerequisites

- WFC Voice Client is loaded on the device and the user is signed in.
- All versions of WFC Voice Client 9.0 support these intents.

Intent Definition

Name	Description
Action	wfc.voice.SIGN_OUT
Intent Type	broadcast
Extra 0	This extra parameter specifies the requested behavior when using the voice client alone or in combination with the Extension Manager.
Type	String
Name	change
Value	Reload: change parameter is not present or change =2
	Change Department: change=1
	Add Department:change=3
	Sign Out: change=0

ADB Examples

Reload:

```
$ adb shell am broadcast -a wfc.voice.SIGN_OUT
```

Change Department:

```
$ adb shell am broadcast -a wfc.voice.SIGN_OUT --es change 1
```

Add Department:

```
$ adb shell am broadcast -a wfc.voice.SIGN_OUT --es change 3
```

Sign Out:

```
$ adb shell am broadcast -a wfc.voice.SIGN_OUT --es change 0
```

Quit

Use Quit to sign out, de-register from the PBX, and quit all application services.

Prerequisites

- WFC Voice Client is loaded on the device.
- All versions of WFC Voice Client 9.0 support these intents.

Intent Definition

Name	Description
Action	wfc.voice.STOP_APP

Actions

Name	Description
Intent Type	broadcast

ADB Example

```
$ adb shell am broadcast -a wfc.voice.STOP_APP
```

Make a Call

There are several intents which can be used to make a voice call. The different intents are supported due to the differing requirements third-party applications have, but they all result in initiating a call in WFC Voice.

Emergency call is supported by all the intents in this section by ensuring that a call can be made at all times, even if the device is locked.

The intents can also be used to bring up the dialpad by omitting the number or address of the intended recipient. The regular dialer is brought up when the device is unlocked; the emergency dialer is brought up when the device is locked.

Supported Schema

The following table summarizes the intents and the supported schemas for each.

Intent	Supported Schemas
wfc.voice.ACTION_BUTTON	"tel", "sip", "csip", "sips"
android.intent.action.VIEW	"tel", "sip", "csip", "sips"
android.intent.action.CALL	"tel", "sip", "sips"
android.intent.action.DIAL	"tel", "sip", "sips"

Prerequisites

- WFC Voice Client is loaded on the device, configured for PBX connectivity, and registered to the PBX.
- All versions of WFC Voice Client 9.0 support these intents.

Make a Call with wfc.voice.ACTION_BUTTON

Name	Description
Action	wfc.voice.ACTION_BUTTON
Intent Type	startActivity
Data Scheme	tel, sip, csip, or sips
Extra 0	
Type	String
Name	action
Value	CALL

Actions

Name	Description
Extra 1	
Type	String
Name	value
Value	Number or address of the intended recipient

Make a Call with `android.intent.action.VIEW`

Name	Value
Action	<code>android.intent.action.VIEW</code>
Intent Type	<code>startActivity</code>
Data Scheme	tel, sip, csip, or sips followed by ":" and the number or address of the intended recipient

Make a Call with `android.intent.action.DIAL`

Name	Description
Action	<code>android.intent.action.CALL</code>
Intent Type	<code>startActivity</code>
Data Scheme	tel, sip, or sips followed by ":" and the number or address of the intended recipient.



NOTE: WFC Voice can be registered to multiple PBX lines simultaneously. This intent is sent to the current line, which can be selected via the WFC Voice dialpad. The user can switch between lines by clicking "Local line" info field.

ADB Examples

```
$ adb shell am start -a wfc.voice.ACTION_BUTTON -d 'tel:' --es action CALL--  
es value 8471234567$
```

```
$ adb shell am start -a android.intent.action.CALL -d sip:2001
```

```
$ adb shell am start -a android.intent.action.VIEW -d csip:2001
```

```
$ adb shell am start -a android.intent.action.DIAL -d tel:555
```

The following intent brings the regular dialer up when the device is unlocked and the emergency dialer when the device is locked.

```
adb shell am start -a android.intent.action.DIAL -d sip:
```

End a Call

Use the End Call action to end an active call session.

Prerequisites

- WFC Voice Client is loaded on the device and configured for PBX connectivity.
- WFC Voice Client is currently involved in a call.
- The minimum required WFC Voice Client version is 9.0.20302.

Intent Definition

Name	Description
Action	<code>com.zebra.wfc.ACTION_END_ACTIVE_SESSION</code>
Intent Type	<code>broadcast</code>

ADB Example

```
$ adb shell am broadcast -a com.zebra.wfc.ACTION_END_ACTIVE_SESSION
```

Import Contacts

Import contacts from a CSV file. The file is stored on the device or on a server.

Prerequisites

- The WFC Voice Client is loaded to the device.
- The CSV file is stored on the device or on a server.
- All versions of WFC Voice Client 9.0 support these intents.

Intent Definition

Name	Description
Action	<code>wfc.voice.SYNC_CONTACTS</code>
Intent Type	<code>startActivity</code>
Extra 0	Location of the contacts CSV file
Type	<code>String</code>
Name	<code>contacts_uri</code>
Value	URL of a downloadable CSV file or a file name on the SD card
Extra 1	This extra parameter specifies if the existing contacts should be deleted or not before applying the CSV file
Type	<code>boolean</code>
Name	<code>reset</code>

Name	Description
Value	<p><code>true</code>: the app deletes all existing contacts prior to loading the CSV file</p> <p><code>false</code>: existing contacts are not deleted</p>

ADB Examples

To load a file stored on the device, and clear existing contacts before applying the file:

```
$ adb shell am start -a wfc.voice.SYNC_CONTACTS --es contacts_uri /sdcard/contacts.csv --ez reset true
```

To load a file stored on a network, and leave existing contacts:

```
$ adb shell am start -a wfc.voice.SYNC_CONTACTS --es contacts_uri https://example.com/voice/contacts/1234 --ez reset false
```

Import Contacts Template

This section provides a description and example of the CSV file template used to import contacts.

- The following fields are required:
 - `contactId`—Use a unique integer for each contact. If a contact already exists with the same `contactId`, the existing record is updated. To delete a contact, set this to a value less than zero.
 - `firstName`—First name
 - `lastName`—Last name
- Provide at least one of the following:
 - `cellNumber`—Cell phone number
 - `officeNumber`—Office phone number
 - `homeNumber`—Home phone number

Optional Fields

- `group`—List group names separated by semicolons.
- `ringtone`—Provide a ringtone for the contact using one of the following value types:
 - `Name`—Name of an existing Android ringtone
 - `URL`—Link to the downloadable music file
 - `Filename`—Path to the music file on the SD card
 - Empty string—Leave blank to use the default ringtone.
- `photo`—URL to a downloadable image file or the path to a file on the SD card. Supported image formats are PNG and JPG.

Table 1 Sample Import

group	contact Id	first Name	last Name	cell Number	office Number	home Number	ringtone	photo
Work; Home; Paint	1	Mike	Smith	2725	847-123- 4560	6540		/sdcard/WFConnect / pic1.jpg
Work; Home	2	John	Page	2726		6541	sample.com / ringtone25.ogg	/sdcard/WFConnect / pic2.jpg
Paint	3	Frank	Flyer	2727			/sdcard/WFConnect / ring4.ogg	sample.com / photo.jpg
Electronics	4	Emilia	Kratzer	2728		6542		/sdcard/WFConnect / pic4.jpg
	-5	Kate	Perry	2729				



NOTE: The column headings in the import CSV file must be one word. For example, contact Id should be contactId. The spaces in the sample are to format the table.

Phone State Monitoring

The WFC Voice Client broadcasts its status using the following intent. This is an intent that is sent by the WFC Voice Client. A third-party application can register to receive the intents; example code is below:

Prerequisites

- WFC Voice Client is loaded to the device.
- The third-party application must register to receive the intents as described in Phone State Example.
- All versions of WFC Voice Client 9.0 support these intents.

Intent Definition

Name	Description
Action	wfc.voice.PHONE_STATE
extras	
registration_state	PBX registration state ACTIVE ACTIVE_DND CONNECTING INACTIVE
state	Voice call state IDLE CALLING RINGING ACTIVE
number	Phone number for the current session that is reported when the voice call state changes (optional).
line_id	Line number that is reported when one of the line registers (optional).
line_extension	Line extension that is reported when one of the lines registers (optional).
line_registered	Boolean expression that is reported when one of the lines registers (optional).

Name	Description
suspended	Boolean expression that is reported when the state is IDLE because one of the sessions is suspended or on hold (optional)



NOTE: The suspended extra is only available in WFC Voice version 9.0.21104 or later.

Phone State Example

The following code example registers the phone state intent from a third party app.

```
// create broadcast receiver
BroadcastReceiver mMMessageReceiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        Log.i(TAG,
            "Received PHONE_STATE from WFCVoice "
            + " registration_state=" +
intent.getStringExtra("registration_state")
            + " call state=" + intent.getStringExtra("state")
            + " number=" + intent.getStringExtra("number")
            + " line_id=" + intent.getStringExtra("line_id")
            + " line_extension=" +
intent.getStringExtra("line_extension")
            + " line_registered=" +
intent.getBooleanExtra("line_registered", false)
            );
    }
};

// register broadcast receiver in the Activity
IntentFilter mMMessageReceiver = new IntentFilter();
requestFilter.addAction("wfc.voice.PHONE_STATE");
registerReceiver(mMMessageReceiver, requestFilter);
```


Headless Mode

Configure the WFC Voice as a background service (headless mode) to allow third-party apps to control voice calls remotely and display the call status on their own user interface (UI) screens.

When headless mode is enabled:

- The WFC Voice home dashboard screen is hidden. Launching the WFC Voice app displays the dial pad or contacts screen.
- Access WFC Voice settings by swiping down from the top of the screen and touching the WFC Voice notification.

Enabling Headless Mode

You can enable headless mode in the XML configuration file or in the app.

- In the WFCConnect XML configuration file, set the `headless_mode` parameter to `true`.
- From the WFC Voice app, touch the three-line menu at the top of the screen and go to **Settings > Advanced Settings > UI Settings > Headless Mode**.

WFC Voice Connector

WFCVoiceConnector is an Android library that allows third-party apps to establish a service connection with a voice app running in headless mode.

The `WFCVoiceConnector.aar` library is part of WFCDemo project.

WFCDemo

WFCDemo is a sample project for an Android app that uses the WFCVoiceConnector library. The app has a UI screen that allows a user to initiate and control voice calls remotely through the service connector project.

The WFCDemo project is available at github.com/ZebraDevs/wfc-voice-demo/.

