

# Deploying and Managing the Applications on Yealink Smart Business Phones

## Introduction

Yealink T5 series smart business phones offer access to a large amount of third-party Android applications. The third-party applications can be installed or uninstalled either one by one or in batch.

There are two versions of the application deployment and management function: special version and standard version. This guide provides detailed information on how to deploy and manage the applications in both two versions. And the features described in this guide are available on the SIP-T58A phones.

## Special Version

In special version, the IP phone runs with highest privileges, you can deploy the applications directly without granting the appropriate permissions to the application. We recommend that you to use the special version with caution, because in this version the IP phone has all the permissions and there may be a security risk.

### To deploy the third-party applications in special version using configuration files:

1. Add/Edit corresponding parameters in configuration files (e.g., app.cfg).

The following table lists the information of parameters:

Parameters	Permitted Values	Default
<b>pm.version</b>	<b>standard, special</b>	<b>standard</b>
<p><b>Description:</b> Configures the version of the application deployment and management function.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.install_url</b>	<b>URL within 511 characters</b>	<b>Blank</b>

Parameters	Permitted Values	Default
<p><b>Description:</b> Configures the access URL of the third-party application.</p> <p><b>Example:</b> app.install_url = http://192.168.10.25/QQ.apk</p> <p>During the auto provisioning process, the IP phone connects to the HTTP provisioning server "192.168.10.25", and downloads the application "QQ.apk".</p> <p>If you want to install/update multiple applications to the phone simultaneously, you can configure as following: app.install_url = http://192.168.10.25/QQ.apk app.install_url = http://192.168.10.25/Kugou.apk</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.uninstall</b>	<b>app name/app name.apk/package name</b>	<b>Blank</b>
<p><b>Description:</b> Uninstalls the specified third-party application.</p> <p><b>Example:</b> To uninstall QQ, do one of the following: app.uninstall = QQ app.uninstall = QQ.apk app.uninstall = com.tencent.mobileqq</p> <p>If you want to uninstall multiple applications on the phone simultaneously, you can configure as following: app.uninstall = QQ app.uninstall = Kugou</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		

The following shows an example of QQ installation in configuration files:

```
pm.version = special
app.install_url = http://192.168.10.25/qq.apk
```

The following shows an example of uninstalling QQ in configuration files:

```
pm.version = special
app.uninstall = qq
```

2. Reference the configuration file in the boot file (e.g., y000000000000.boot).

Example:

```
include:config "http://192.168.10.25/app.cfg"
```

3. Upload the boot file and configuration file to the root directory of the provisioning server.
4. Trigger IP phones to perform an auto provisioning for configuration update.

For more information on auto provisioning, refer to the latest Auto Provisioning Guide on [Yealink Technical Support](#).

## Standard Version

In standard version, the IP phone does not have all the permissions, only the administrator can deploy and manage applications using configuration files. There are a series of parameters for administrator to control the applications and grant permissions to applications.

After installing the application, you may need to grant the appropriate permissions to the application to use the application.

## Deploying the Third-party Applications

**To deploy the third-party applications in standard version using configuration files:**

1. Add/Edit corresponding parameters in configuration files (e.g., app.cfg).

The following table lists the information of parameters:

Parameters	Permitted Values	Default
<b>pm.version</b>	<b>standard, special</b>	<b>standard</b>
<p><b>Description:</b> Configures the version of the application deployment and management function.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		

Parameters	Permitted Values	Default
<b>app.install_url</b>	<b>URL within 511 characters</b>	<b>Blank</b>
<p><b>Description:</b> Configures the access URL of the third-party application.</p> <p><b>Example:</b> app.install_url = http://192.168.10.25/QQ.apk</p> <p>During the auto provisioning process, the IP phone connects to the HTTP provisioning server "192.168.10.25", and downloads the application "QQ.apk".</p> <p>If you want to install/update multiple applications to the phone simultaneously, you can configure as following: app.install_url = http://192.168.10.25/QQ.apk app.install_url = http://192.168.10.25/Kugou.apk</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.uninstall</b>	<b>app name/app name.apk/package name</b>	<b>Blank</b>
<p><b>Description:</b> Uninstalls the specified third-party application.</p> <p><b>Example:</b> To uninstall QQ, do one of the following: app.uninstall = QQ app.uninstall = QQ.apk app.uninstall = com.tencent.mobileqq</p> <p>If you want to uninstall multiple applications on the phone simultaneously, you can configure as following: app.uninstall = http://192.168.10.25/QQ.apk app.uninstall = http://192.168.10.25/Kugou.apk</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		

The following shows an example of QQ installation in configuration files:

```
pm.version = standard
app.install_url = http://192.168.10.25/qq.apk
```

The following shows an example of uninstalling QQ in configuration files:

```
pm.version = standard
app.uninstall = qq
```

2. Reference the configuration file in the boot file (e.g., y000000000000.boot).

Example:

```
include:config "http://192.168.10.25/app.cfg"
```

3. Upload the boot file and configuration file to the root directory of the provisioning server.
4. Trigger IP phones to perform an auto provisioning for configuration update.

For more information on auto provisioning, refer to the latest Auto Provisioning Guide on [Yealink Technical Support](#).

In order to use the application, you may need to grant appropriate permissions to the application. For more information, refer to [Permission Control](#) on page 8.

You can also control the application to run after startup, run in the background, run during the call and so on. For more information, refer to [Application Control](#) on page 5.

## Managing the Applications

### Application Control

**To control the applications using configuration files:**

1. Add/Edit corresponding parameters in configuration files (e.g., app.cfg).

The following table lists the information of parameters:

Parameters	Permitted Values	Default
<b>app.unavailable</b>	<b>app name/app name.apk/package name</b>	<b>none</b>
<p><b>Description:</b> Hides the system applications on the IP phone.</p> <p><b>Example:</b> app.unavailable = Settings</p> <p>If you want to hide multiple applications on the phone simultaneously, you can configure as following (between two applications are separated by "/"):  app.unavailable = Settings/FileManager/Message</p>		

Parameters	Permitted Values	Default
<p>If you want to cancel the operations for applications on the phone, you can configure as following:</p> <p>app.unavailable = none</p> <p><b>Note:</b> It works only if the value of the parameter "pm.version" is set to standard. It is only applicable to the system applications.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.autorun</b>	<b>app name/app name.apk/package name</b>	<b>none</b>
<p><b>Description:</b> Allows the application to run automatically after the phone starts up. For more information on how to configure it, refer to the parameter "app.unavailable".</p> <p><b>Note:</b> It works only if the value of the parameter "pm.version" is set to standard.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.prohibit_bgrun</b>	<b>app name/app name.apk/package name</b>	<b>none</b>
<p><b>Description:</b> Prohibits the application to run in the background. For more information on how to configure it, refer to the parameter "app.unavailable".</p> <p><b>Note:</b> It works only if the value of the parameter "pm.version" is set to standard.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.prohibit_callrun</b>	<b>app name/app name.apk/package name</b>	<b>none</b>

Parameters	Permitted Values	Default
<p><b>Description:</b> Prohibits the application to run during the call. For more information on how to configure it, refer to the parameter "app.unavailable".</p> <p><b>Note:</b> It works only if the value of the parameter "pm.version" is set to standard.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		
<b>app.prohibit_uninstall</b>	<b>app name/app name.apk/package name</b>	<b>none</b>
<p><b>Description:</b> Prohibits the users to uninstall the third-party application via phone user interface. For more information on how to configure it, refer to the parameter "app.unavailable".</p> <p><b>Note:</b> It works only if the value of the parameter "pm.version" is set to standard. It is only applicable to the third-party applications.</p> <p><b>Web User Interface:</b> None</p> <p><b>Phone User Interface:</b> None</p>		

The following shows an example of QQ installation in configuration files:

```
app.unavailable = Settings
app.autorun = QQ
app.prohibit_bgrun = Recorder
app.prohibit_callrun = Browser
app.prohibit_uninstall = QQ
```

- Reference the configuration file in the boot file (e.g., y000000000000.boot).

Example:

```
include:config "http://192.168.10.25/app.cfg"
```

- Upload the boot file and configuration file to the root directory of the provisioning server.
- Trigger IP phones to perform an auto provisioning for configuration update.

For more information on auto provisioning, refer to the latest Auto Provisioning Guide on [Yealink Technical Support](#).

## Permission Control

Yealink provide a series of parameters that allow you to manage the third-party application permissions. With the help of these parameters, you can easily grant the privacy data/security control/system tool permissions to the third-party applications.

### To grant permissions to applications in standard version using configuration files:

1. Add/Edit corresponding parameters in configuration files (e.g., app.cfg).

The following table lists the information of parameters:

Parameters	Permitted Values		Description
<b>app.permission</b>	all	Grants the permission to all third-party applications.	Allows a third-party application to access all limited-level permissions described in following.
	app name/ app name.apk/ package name	Grants the permission to the specified third-party application.	
		Grants the permission to multiple third-party applications simultaneously (between two applications are separated by "/").	
	none	Cancels the operations for third-party applications on the phone.	
<b>Privacy Data:</b>			



Parameters	Permitted Values	Description
<b>app.permission.privacy_data</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to access the following privacy data.
<b>app.permission.privacy_data.aaccount</b>		Allows a third-party application to direct access to the accounts managed by the Account Manager.
<b>app.permission.privacy_data.aaccount.get</b>		Allows a third-party application to access the list of accounts in the Accounts Service.
<b>app.permission.privacy_data.aaccount.auth</b>		Allows a third-party application to act as an AccountAuthenticator for the AccountManager.
<b>app.permission.privacy_data.aaccount.manage</b>		Allows a third-party application to manage the list of accounts in the AccountManager.
<b>app.permission.privacy_data.aaccount.credential</b>		Allows a third-party application to request authentication from the AccountManager.
<b>app.permission.privacy_data.location</b>		Allows a third-party application to access the following location information.
<b>app.permission.privacy_data.location.coarse</b>		Allows a third-party application to access approximate location derived from network location sources such as cell towers and Wi-Fi.
<b>app.permission.privacy_data.location.fine</b>		Allows a third-party application to access precise location from location sources such as GPS, cell towers, and Wi-Fi.

Parameters	Permitted Values	Description
<b>app.permission.privacy_data.location.mock</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to create mock location providers for testing.
<b>app.permission.privacy_data.location.extra</b>		Allows a third-party application to access extra location provider commands.
<b>app.permission.privacy_data.phone_state</b>		Allows a third-party application to access phone state.
<b>app.permission.privacy_data.bookmarks</b>		Allows a third-party application to read and write the user's browsing history and bookmarks.
<b>app.permission.privacy_data.bookmarks.read</b>		Allows a third-party application to read the user's browsing history and bookmarks.
<b>app.permission.privacy_data.bookmarks.write</b>		Allows a third-party application to write (but not read) the user's browsing history and bookmarks.
<b>app.permission.privacy_data.calendar</b>		Allows a third-party application to read and write the user's calendar data.
<b>app.permission.privacy_data.calendar.read</b>		Allows a third-party application to read the user's calendar data.
<b>app.permission.privacy_data.calendar.write</b>		Allows a third-party application to write (but not read) the user's calendar data.
<b>app.permission.privacy_data.profile</b>		Allows a third-party application to read and write the user's personal profile data.

Parameters	Permitted Values	Description
<b>app.permission.privacy_data.profile.read</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to read the user's personal profile data.
<b>app.permission.privacy_data.profile.write</b>		Allows a third-party application to write (but not read) the user's personal profile data.
<b>app.permission.privacy_data.call_log</b>		Allows a third-party application to read and write the user's call log.
<b>app.permission.privacy_data.call_log.read</b>		Allows a third-party application to read the user's call log.
<b>app.permission.privacy_data.call_log.write</b>		Allows a third-party application to write (but not read) the user's call log.
<b>app.permission.privacy_data.contacts</b>		Allows a third-party application to read and write the user's contacts data.
<b>app.permission.privacy_data.contacts.read</b>		Allows a third-party application to read the user's contacts data.
<b>app.permission.privacy_data.contacts.write</b>		Allows a third-party application to write (but not read) the user's contacts data.
<b>app.permission.privacy_data.subscribed_feed</b>		Allows a third-party application to read and write the subscribed feeds ContentProvider.
<b>app.permission.privacy_data.subscribed_feed.read</b>		Allows a third-party application to read the subscribed feeds ContentProvider.

Parameters	Permitted Values	Description
<b>app.permission.privacy_data.subscribed_feed.write</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to write (but not read) the subscribed feeds ContentProvider.
<b>app.permission.privacy_data.user_dictionary</b>		Allows a third-party application to read and write the user dictionary.
<b>app.permission.privacy_data.user_dictionary.read</b>		Allows a third-party application to read the user dictionary.
<b>app.permission.privacy_data.user_dictionary.write</b>		Allows a third-party application to write (but not read) to the user dictionary.
<b>app.permission.privacy_data.voicemail</b>		Allows a third-party application to add voicemails into the system.
<b>Security Control:</b>		
<b>app.permission.security_control</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to access the following security control.
<b>app.permission.security_control.camera</b>		Allows a third-party application to access the camera device.
<b>app.permission.security_control.microphone</b>		Allows a third-party application to record audio.
<b>app.permission.security_control.screen_lock</b>		Allows a third-party application to disable the phone lock.

Parameters	Permitted Values	Description
<b>app.permission.security_control.get_tasks</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to access the current or recently used applications list.
<b>app.permission.security_control.show_alert_window</b>		Allows a third-party application to open windows using the type TYPE_SYSTEM_ALERT, shown on top of all other applications.
<b>app.permission.security_control.clear_app_cache</b>		Allows a third-party application to clear the caches of all installed applications on the IP phone.
<b>app.permission.security_control.shortcut</b>		Allows a third-party application to install or uninstall a shortcut in Launcher.
<b>app.permission.security_control.shortcut.install</b>		Allows a third-party application to install a shortcut in Launcher.
<b>app.permission.security_control.shortcut.uninstall</b>		Allows a third-party application to uninstall a shortcut in Launcher.
<b>System Tool:</b>		
<b>app.permission.system_tool</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to use the following system tool.
<b>app.permission.system_tool.kill_background_processes</b>		Allows a third-party application to call killBackgroundProcesses(String).
<b>app.permission.system_tool.get_package_size</b>		Allows a third-party application to find out the space used by any package.

Parameters	Permitted Values	Description
<b>app.permission.system_tool.receive_boot_completed</b>	For more information on how to configure it, refer to the parameter "app.permission".	Allows a third-party application to receive the ACTION_BOOT_COMPLETED that is broadcast after the system finishes booting.
<b>app.permission.system_tool.broadcast_sticky</b>		Allows a third-party application to broadcast sticky intents.
<b>app.permission.system_tool.expand_status_bar</b>		Allows a third-party application to expand or collapse the control center and notification center.
<b>app.permission.system_tool.audio_setting</b>		Allows a third-party application to modify global audio settings.
<b>app.permission.system_tool.system_setting</b>		Allows a third-party application to read and write the system settings.
<b>app.permission.system_tool.sync_setting</b>		Allows a third-party application to read and write the sync settings.
<b>app.permission.system_tool.sync_setting.read</b>		Allows a third-party application to read the sync settings.
<b>app.permission.system_tool.sync_setting.write</b>		Allows a third-party application to write the sync settings.
<b>app.permission.system_tool.sync_setting.read_stats</b>		Allows a third-party application to read the sync status.

2. Reference the configuration file in the boot file (e.g., y000000000000.boot).

Example:

```
include:config "http://192.168.10.25/app.cfg"
```

3. Upload the boot file and configuration file to the root directory of the provisioning server.
4. Trigger IP phones to perform an auto provisioning for configuration update.

For more information on auto provisioning, refer to the latest Auto Provisioning Guide on [Yealink Technical Support](#).

## Customer Feedback

We are striving to improve our documentation quality and we appreciate your feedback. Email your opinions and comments to [DocsFeedback@yealink.com](mailto:DocsFeedback@yealink.com).