

ADVANCED MOBILE PAYMENT INC.

ATMS AGENT LIBRARY

Developer Guide

v 1.4

PTM- 0210-1.4-E

This information is CONFIDENTIAL and must be used exclusively for the operation of AMP POS by the Advanced Mobile Payment Inc. It may not be duplicated, published, or disclosed without written permission.



PROPRIETARY NOTICE

All pages of this document contain information proprietary to Advanced Mobile Payment (AMP) Inc. This document shall not be duplicated, transmitted, used, or otherwise disclosed to anyone other than the organization or specific individuals to which this document is delivered. This restriction is applicable to all sheets of this document. AMP reserves the right to have the recipient return all copies of this document at any time. AMP reserves the right to change the content of this document without prior notice.

© 2021 AMP Inc. All Rights Reserved.

DOCUMENT PROPERTIES

INFORMATION

ID	PTM- 0210-1.3-E
Title	ATMS Agent Library - Developer Guide
Document Portal Path	/Development/AMP POS/AMP 6,8 Series/ATMS Agent Library
Category	AMP POS – TMS
Access Level	Development
NDA Required	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>

VERSION CONTROL

Template Version	1.5
-------------------------	-----

Doc. Version	Date	Summary of Change	Updated by
1.0	Jun 2020	Initial version	B.Zhang
1.1	Aug 2020	Added the api that were added to ATMS Agent Lib after v1.4.0	B.Zhang
1.2	Nov 2020	Added: <ul style="list-style-type: none"> deleteInternalStorage() and callback getAuthCode() and callback 	B.Zhang
1.3	Apr 2021	Added: <ul style="list-style-type: none"> getApplicationParameters() and callbacks setApplicationParameter() and callbacks Deprecated setConfigurationParameter() and the corresponding callback	B.Zhang
1.4	Jun 2021	Updated: <ul style="list-style-type: none"> AMP360ServerAPIUrl v2 encryption api from desDecryptUnifiedByCUID -> desDecryptUnified Deprecated api 'getAuthCode()' 	B.Zhang

SUPPORTED HARDWARE & SOFTWARE

Doc. Version	Software Title	Release	Supported Hardware Model
1.0	ATMS Agent Lib	V1.4.0	AMP 6 and 8 Series
1.0	ATMS Agent	V2.5.1	AMP 6 and 8 Series

1.0	Firmware	V3.2.23	AMP 8 Series
1.0	Firmware	V1.0.14	AMP 6500
1.0	Firmware	V6.2.5	AMP 6700
1.1	ATMS Agent Lib	V1.8.0	AMP 6 and 8 Series
1.1	ATMS Agent	V2.9.0	AMP 6 and 8 Series
1.1	Firmware	V3.2.23	AMP 8 Series
1.1	Firmware	V1.0.14	AMP 6500
1.2	ATMS Agent Lib	V1.8.1	AMP 8 Series, AMP 6500
1.2	ATMS Agent	V2.10.0	AMP 8 Series, AMP 6500
1.3	ATMS Agent Lib	V1.8.4	AMP 8 Series, AMP 6500
1.3	Firmware	V1.0.20	AMP 6500
1.3 – 1.4	ATMS Agent	V2.11.1	AMP 8 Series, AMP 6500
1.4	ATMS Agent Lib	V1.9.0	AMP 8 Series, AMP 6500
1.4	Firmware	V1.0.22	AMP 6500
1.4	Firmware	V3.2.39	AMP 8 Series

EXTERNAL REFERENCES

URLs	https://api.amp360.amobilepayment.com/AMP360-api https://dev.api.amp360.amobilepayment.com/AMP360-api
E-Mail Addresses	--
Phone Numbers	--
Documents	--
Knowledge Base Articles	--

TABLE OF CONTENTS

Proprietary Notice	i
Document Properties	ii
Table of Contents	iv
Table of Figures	vi
1 AMP Android Terminal ATMS Agent Library	1
2 Functions	1
2.1 init	1
2.2 checkForUpdates	1
2.3 cancelCheckForUpdates	2
2.4 installApplication	2
2.5 installApplication with Configuration	3
2.6 installFirmware	3
2.7 silentInstallAllDownloads	4
2.8 getConfigurationParameters	5
2.9 getConfigurationParametersHash	5
2.10 getAdditionalFile	6
2.11 getAdditionalFileHash	6
2.12 getLocation	7
2.13 uploadFile	7
2.14 setATMSServerAddress	8
2.15 setTerminalID	8
2.16 setCellularDownload	8
2.17 setApplicationParameter	9
2.18 setApplicationParameter	10
2.19 getApplicationParameters	10
2.20 getApplicationParameters	11
2.21 deleteInternalStorage	12
2.22 (Deprecated)getAuthCode	12

2.23	(Deprecated)setConfigurationParameter.....	12
2.24	(Deprecated) updateFirmware	13
3	Callbacks	14
3.1	onJobError	14
3.2	onJobCompleted	14
3.3	onDownloadStatus	14
3.4	onDownloadProgress	15
3.5	onDownloadJobNumber	15
3.6	onFirmwareUpgradeStatus.....	15
3.7	onKeyInjectionReturnCode	16
3.8	onRequestedFileNotFound	16
3.9	onAppInstallCompleted.....	16
3.10	onAdditionalFilesReceived.....	17
3.11	onAdditionalFilesHashReceived	17
3.12	onConfigurationParametersReceived	17
3.13	onConfigurationParametersHashReceived	17
3.14	onSilentInstallAllLastDownloadsFinished	18
3.15	onLocationDetected	18
3.16	onCellularDownloadSet	19
3.17	onDeleteInternalStorageFinished	19
3.18	onSetApplicationParameterResponse.....	19
3.19	onSetApplicationParameterFail	20
3.20	onGetApplicationParametersResponse	20
3.21	onGetApplicationParametersFail	20
3.22	(Deprecated)onSetConfigurationParameterStatus	21

TABLE OF FIGURES

No table of figures entries found.

1 AMP ANDROID TERMINAL ATMS AGENT LIBRARY

AMP 360 is the complete Terminal Management System which enables the efficient management of high cost mass updates to the application and parameters within the AMP terminals. It facilitates file downloads along with terminal parameters in a multi-application environment. AMP 360 can be accessed through the application's user interface and/or the API level on an AMP terminal.

An interface is required for enabling other applications, that run on AMP terminals or sometimes on other Android devices, to benefit from the previously noted features. The ATMS Agent Library is used by these applications to communicate with the AMP 360.

This document details out the ATMS Agent Library functions and callbacks. The ATMS Agent Library functions rely on AMP terminal's ATMS Agent application and firmware and therefore are exclusive to the applications that run on AMP terminals.

2 FUNCTIONS

2.1 INIT

Signature	<code>public void init(Context context,IServerResponse iServerResponse)</code>
Description	This function should be called before any other function. It initialises the SDK library and broadcasts receivers. This function should only be called once.
Parameters	<ul style="list-style-type: none"> context: the context of the single, global Application object of the current process. iServerResponse: the callback object that contains the task's information status.
Callbacks	<ul style="list-style-type: none"> onKeyInjectionReturnCode: This is always called.
Notes	<ul style="list-style-type: none"> Supported since v1.4.0. Exclusive to AMP terminals

2.2 CHECKFORUPDATES

Signature	<code>public void checkForUpdates()</code>
Description	This function inspects the availability of any new firmware updates, applications, configuration parameters or any additional files.

Parameters	None
Callbacks	<ul style="list-style-type: none"> • onDownloadStatus: This is called when the current file download status is changed. • onDownloadProgress: This is called when the current file is being downloaded. It notifies the user of the download process. • onDownloadJobNumber: This is called when each file starts downloading. • onJobError: This is called when any error happens during the process. • onJobCompleted: This is called when the download job is completed without errors.
Notes	<ul style="list-style-type: none"> • Supported since v1.4.0. • Exclusive to AMP terminals

2.3 CANCELCHECKFORUPDATES

Signature	<code>public void cancelCheckForUpdates()</code>
Description	This function cancels the download process.
Parameters	None.
Callbacks	None.
Notes	<ul style="list-style-type: none"> • Supported since v1.4.0. • Exclusive to AMP terminals

2.4 INSTALLAPPLICATION

Signature	<code>public void installApplication()</code>
Description	This function initiates the installation process of the APK file which corresponds to the application calling the function.
Parameters	None.
Callbacks	<ul style="list-style-type: none"> • onRequestedFileNotFound: This is called when the requested apk file is not found. The TMS_FILE_TYPE parameter value will be APPLICATION. • onAppInstallCompleted: This is called when the requested apk file is found

Notes	<ul style="list-style-type: none"> Supported since v1.4.0. onAppInstallCompleted is supported since v1.8.0 Exclusive to AMP terminals
--------------	--

2.5 INSTALLAPPLICATION WITH CONFIGURATION

Signature	<code>public void installApplication(Constants.INSTALL_MODE installMode)</code>
Description	<p>This function initiates the installation process of the APK file which corresponds to the application that is calling the function.</p> <p>When silent installation is enabled, the apk will be silently installed. There will be no UI changes and no user intervention will be required. When auto run is enabled, the app will be automatically launched upon installation completion.</p>
Parameters	<p>Constants.INSTALL_MODE</p> <ul style="list-style-type: none"> MODE_NORMAL_INSTALL Standard way to install an apk, the Android installation page will be displayed. MODE_SILENT_INSTALL Silent installation of the apk without user intervention. MODE_AUTO_RUN_AFTER_SILENT_INSTALLED The app will be automatically launched after installation completion.
Callbacks	<p>onRequestedFileNotFound: This is called when the requested apk file is not found. The TMS_FILE_TYPE parameter value will be APPLICATION.</p> <ul style="list-style-type: none"> onAppInstallCompleted: This is called when the requested apk file is found.
Notes	<ul style="list-style-type: none"> Supported since v1.8.0. Exclusive to AMP terminals

2.6 INSTALLFIRMWARE

Signature	<code>public void installFirmware()</code>
Description	This function starts the firmware update process when an updated firmware file is available.
Parameters	None.

Callbacks	<ul style="list-style-type: none"> • <code>onRequestedFileNotFound</code>: This is called when the requested firmware file is not found. The <code>TMS_FILE_TYPE</code> parameter value will be <code>FIRMWARE</code>. • <code>onFirmwareUpgradeStatus</code>: This is called when the firmware image file is found. Note that this is called before the actual firmware upgrade starts.
Notes	<ul style="list-style-type: none"> • Supported since v1.8.0. • Exclusive to AMP terminals

2.7 SILENTINSTALLLASTALLDOWNLOADS

Signature	<code>public void silentInstallLastAllDownloads()</code>
Description	<p>This function silently installs all the applications' apk files and firmware image files that are downloaded in the most recent job download, which is triggered by the most recent <code>checkForUpdate</code> call. Application files are installed first, then the firmware installation happens.</p> <p>When more than one application is installed or when firmware upgrade occurs, the terminal will restart.</p>
Parameters	None.
Callbacks	<ul style="list-style-type: none"> • <code>onSilentInstallAllLastDownloadsFinished</code>: This callback is always called, except for the following scenarios: <ul style="list-style-type: none"> ◦ When firmware update is needed, the firmware update start is unsuccessful and only one apk file is installed, with a package name that matches that of the calling app. The calling app will therefore be relaunched, and the callback will not be called. ◦ When firmware update is not needed and only one apk file is installed, with a package name that matches that of the calling app. The calling app will therefore be relaunched, and the callback will not be called.
Notes	<ul style="list-style-type: none"> • Supported since v1.8.0. • <code>onRequestedFileNotFound()</code> won't be triggered in this case. • Exclusive to AMP terminals

2.8 GETCONFIGURATIONPARAMETERS

Signature	<code>public void getConfigurationParameters()</code>
Description	When this function is called, if the configuration parameters file is available and the package name matches that of the calling app, the function copies them to the corresponding application's internal memory.
Parameters	None.
Callbacks	<ul style="list-style-type: none"> onRequestedFileNotFound: This is called when the configuration parameter file is not found. The TMS_FILE_TYPE parameter value will be CONFIGURATION_PARAMETERS. onConfigurationParametersReceived: This is called when the configuration parameter file is found, and its package name matches that of the calling app. If the package name doesn't match, this won't be called.
Notes	<ul style="list-style-type: none"> Supported since v1.4.0. Exclusive to AMP terminals

2.9 GETCONFIGURATIONPARAMETERSHASH

Signature	<code>public void getConfigurationParametersHash()</code>
Description	This function returns the hash code of the configuration parameter file when it is available.
Parameters	None.
Callbacks	<ul style="list-style-type: none"> onRequestedFileNotFound: This is called when the configuration parameter file is not found, the TMS_FILE_TYPE parameter value will be CONFIGURATION_PARAMETERS onConfigurationParametersReceived: This is called when the configuration parameter file is found, and its package name matches that of the calling app. If the package name does not match, this will not be called.
Notes	<ul style="list-style-type: none"> Supported since v1.8.0. Exclusive to AMP terminals

2.10 GETADDITIONALFILE

Signature	<code>public void getAdditionalFile()</code>
Description	This function is called, if additional files are available and the package name matches that of the calling app, the function copies them to the corresponding application's internal memory.
Parameters	None.
Callbacks	<ul style="list-style-type: none"> onRequestedFileNotFound: This is called when the requested additional file is not found. The TMS_FILE_TYPE parameter value will be ADDITIONAL_FILES. onAdditionalFilesReceived: This is called when the requested additional file is found, and its package name matches that of the calling app. If the package name does not match, this will not be called.
Notes	<ul style="list-style-type: none"> Supported since v1.4.0. Exclusive to AMP terminals

2.11 GETADDITIONALFILEHASH

Signature	<code>public void getAdditionalFileHash()</code>
Description	This function returns the additional file's hash code when it is available.
Parameters	None
Callbacks	<ul style="list-style-type: none"> onRequestedFileNotFound: This is called when the requested additional file is not found. The TMS_FILE_TYPE parameter value will be ADDITIONAL_FILES. onAdditionalFilesReceived: This is called when the requested additional file is found, and its package name matches that of the calling app. If the package name does not match, this will not be called.
Notes	<ul style="list-style-type: none"> Supported since v1.8.0 Exclusive to AMP terminals

2.12 GETLOCATION

Signature	<code>public void getLocation()</code>
Description	This function will get the location information saved in the ATMSAgent.
Parameters	None.
Callbacks	<ul style="list-style-type: none"> onLocationDetected: This is always called. If the location info is unavailable, an empty text will be returned.
Notes	<ul style="list-style-type: none"> Supported since v1.8.0. The response location data is not in JSON format. Exclusive to AMP terminals

2.13 UPLOADFILE

Signature	<code>public void uploadFile(final File file)</code>
Description	This function uploads a file from the application's internal memory to the TMS server. The process is the same as checkForUpdate()
Parameters	<ul style="list-style-type: none"> file: Upload file instance
Callbacks	<ul style="list-style-type: none"> onDownloadStatus: This is called when the current download file status is changed. onDownloadProgress: This is called when the current file is downloading and notifies the user of the download process. onDownloadJobNumber: This is called when each file starts downloading. onJobError: This is called when an error happens during the process. onJobCompleted: This is called when file upload is completed without error.
Notes	<ul style="list-style-type: none"> Supported since v1.4.0 This function has a limitation, the uploaded file should not be greater than 1024 bytes in size. Exclusive to AMP terminals

2.14 SETATMSSERVERADDRESS

Signature	<code>public void setATMSServerAddress(String aTMSServerAddress)</code>
Description	This function sets the TMS server address in the calling app and updates the TMS server address in the ATMS agent application.
Parameters	<ul style="list-style-type: none"> aTMSServerAddress: The TMS server URL address.
Callbacks	None.
Notes	<ul style="list-style-type: none"> Supported since v1.4.0 Exclusive to AMP terminals

2.15 SETTERMINALID

Signature	<code>public void setTerminalID(String terminalID)</code>
Description	This function sets the terminal ID in the calling app and updates the terminal ID in the ATMS agent application.
Parameters	<ul style="list-style-type: none"> terminalID: The terminal ID.
Callbacks	None.
Notes	<ul style="list-style-type: none"> Supported since v1.4.0. Exclusive to AMP terminals

2.16 SETCELLULARDOWNLOAD

Signature	<code>public void setCellularDownload(boolean flag)</code>
Description	This function will trigger ATMSAgent to update cellular download setting value (true or false) to avoid unnecessary data consumption. When it is set to 'true', cellular data will be used when WiFi and Ethernet are not available.
Parameters	flag: A boolean value to set cellular download true or false
Callbacks	<ul style="list-style-type: none"> onCellularDownloadSet(Boolean cellularDownloadSet): This is always called. The parameter indicates the current value of cellular download.

Notes	<ul style="list-style-type: none"> Supported since v1.8.0. Exclusive to AMP terminals
--------------	---

2.17 SETAPPLICATIONPARAMETER

Signature	<code>public void setApplicationParameter(String url, String packageName, String tag, String value)</code>
Description	<p>This function facilitates the application parameter value upload to AMP 360 server.</p> <p>The recommended use of this function is for testing since it allows the connection to the dev server.</p>
Parameters	<ul style="list-style-type: none"> url: the server URL. This value can point to the dev server, for tests, or the production (default) value. <ul style="list-style-type: none"> The production server URL, or the default value, is <code>https://api.amp360.amobilepayment.com/AMP360-api</code>. The development server URL, to be used for tests is <code>https://dev.api.amp360.amobilepayment.com/AMP360-api</code>. If the value is "", the default URL will be used. packageName: The application's packageName. tag: The parameter's tag field, as is defined in the server. value: The value that the parameter should be set to in the server.
Callbacks	<ul style="list-style-type: none"> <code>onSetApplicationParameterResponse(final AMP360Response amp360Response);</code> This callback is called when server communication is successful. In this case, all server response information can be found in the 'amp360Response' object. <code>onSetApplicationParameterFail(final String errorDesc)</code> This callback is called when server communication fails.
Notes	<ul style="list-style-type: none"> Supported since v1.8.4. Exclusive to AMP terminals.

2.18 SETAPPLICATIONPARAMETER

Signature	<code>public void setApplicationParameter(String packageName, String tag, String value)</code>
Description	This function facilitates the application parameter value upload to AMP 360 server. This is the same as the previous function with one difference. The server URL is not a parameter, and the default or production server URL is used by this function.
Parameters	<ul style="list-style-type: none"> • <code>packageName</code>: The application's <code>packageName</code>. • <code>tag</code>: The parameter's <code>tag</code> field, as is defined in the server. • <code>value</code>: The value that the parameter should be set to in the server.
Callbacks	<ul style="list-style-type: none"> • <code>onSetApplicationParameterResponse(final AMP360Response amp360Response)</code>; This callback is called when server communication is successful. In this case, all server response information can be found in the 'amp360Response' object. • <code>onSetApplicationParameterFail(final String errorDesc)</code> This callback is called when server communication fails.
Notes	<ul style="list-style-type: none"> • Supported since v1.8.4. • Exclusive to AMP terminals.

2.19 GETAPPLICATIONPARAMETERS

Signature	<code>public void getApplicationParameters(String url, String serialNumber, String packageName)</code>
Description	<p>This function returns all the application parameters information from the server.</p> <p>The recommended use of this function is for testing since it allows the connection to the dev server.</p>
Parameters	<ul style="list-style-type: none"> • <code>url</code>: the server URL. This value can point to the dev server, for tests, or the production (default) value. <ul style="list-style-type: none"> ◦ The production server URL, or the default value, is <code>https://api.amp360.amobilepayment.com/AMP360-api</code>. ◦ The development server URL, to be used for tests is <code>https://dev.api.amp360.amobilepayment.com/AMP360-api</code>. ◦ If the value is "", the default URL will be used.

	<ul style="list-style-type: none"> • <code>serialNumber</code>: The terminal's <code>serialNumber</code>. • <code>packageName</code>: The application's <code>packageName</code>.
Callbacks	<ul style="list-style-type: none"> • <code>onGetApplicationParametersResponse</code>(final AMP360Response amp360Response); This callback is called when server communication is successful. In this case, all server response information can be found in the 'amp360Response' object. • <code>onGetApplicationParametersFail</code>(final String errorDesc) This callback is called when server communication fails.
Notes	<ul style="list-style-type: none"> • Supported since v1.8.4. • Exclusive to AMP terminals

2.20 GETAPPLICATIONPARAMETERS

Signature	<code>public void getApplicationParameters(String serialNumber, String packageName)</code>
Description	This function returns all the application parameters information from the server. This is the same as the previous function with one difference. The server URL is not a parameter, and the default or production server URL is used by this function.
Parameters	<ul style="list-style-type: none"> • <code>serialNumber</code>: The terminal's <code>serialNumber</code> • <code>packageName</code>: The application's <code>packageName</code>
Callbacks	<ul style="list-style-type: none"> • <code>onGetApplicationParametersResponse</code>(final AMP360Response amp360Response); This callback is called when server communication is successful. In this case, all server response information can be found in the 'amp360Response' object. • <code>onGetApplicationParametersFail</code>(final String errorDesc) This callback is called when server communication fails.
Notes	<ul style="list-style-type: none"> • Supported since v1.8.4. • Same function as 2.20, except the url is not a required parameter, instead, the default url will be used • Exclusive to AMP terminals

2.21 DELETEINTERNALSTORAGE

Signature	<code>public void deleteInternalStorage()</code>
Description	This function will trigger the ATMS Agent function that deletes all previously downloaded files from ATMS Agent cache and clears the list of the files that were downloaded in the last call to <code>checkForUpdates()</code> .
Parameters	
Callbacks	<ul style="list-style-type: none"> <code>onDeleteInternalStorageFinished(boolean isAllFileDeleted, ArrayList<String> failList)</code>: This is always called. The parameter 'isAllFileDeleted' indicates if all downloaded files are deleted. If not, a list of app names which failed to be deleted is returned in 'failList'.
Notes	<ul style="list-style-type: none"> Supported since v1.8.0. Exclusive to AMP terminals

2.22 (DEPRCATED)GETAUTHCODE

Signature	<code>public void getAuthCode(String packageName)</code>
Description	This function will calculate and return device serial number, challenge, key code and auth code.
Parameters	<code>packageName</code> : A full package name of any app
Callbacks	No callback is called, the result will be returned as JSON String immediately.
Notes	<ul style="list-style-type: none"> Supported since v1.8.0 Deprecated since v1.9.0.

2.23 (DEPRCATED)SETCONFIGURATIONPARAMETER

Signature	<code>public void setConfigurationParameter(String tid, String tag, String value, String packageName)</code>
Description	This function facilitates parameter value upload to server to ensure the parameter value on server and the terminal are the same. This function can be called when the value is changed by the Merchant on the terminal.

Parameters	<ul style="list-style-type: none"> • tid: Terminal serial number • tag: Tag field defined in server • value: Value that matches the tag • packageName: The packageName of the app
Callbacks	<ul style="list-style-type: none"> • onSetConfigurationParameterStatus(int responseCode, String respDesc): This is always called. The parameter will indicate the response code and description received from server.
Notes	<ul style="list-style-type: none"> • Supported since v1.8.0. • Deprecated since v1.8.4, please see the alternative api "setApplicationParameter()" • Exclusive to AMP terminals

2.24 (DEPRECATED) UPDATEFIRMWARE

Signature	<code>public void updateFirmware()</code>
Description	<p>This function is deprecated and should no longer be used. Please see installFirmware instead.</p> <p>Calling this function always results to onRequestedFileNotFound(TMS_FILE_TYPE FIRMWARE) call back.</p>
Parameters	None.
Callbacks	<ul style="list-style-type: none"> • onRequestedFileNotFound(TMS_FILE_TYPE FIRMWARE): This callback is always called.
Notes	<ul style="list-style-type: none"> • Supported since v1.4.0. • This function was deprecated since v1.8.0. • Exclusive to AMP terminals

3 CALLBACKS

3.1 ONJOBERROR

Signature	<code>void onJobError(STATE state, String error, String debugError);</code>
Description	This callback logs any error occurrence.
Parameters	<ul style="list-style-type: none"> • state: indicates the error source. The following is a list of the possible states: <ul style="list-style-type: none"> ◦ SERVER_AUTHENTICATION, ◦ AGENT_VERIFICATION, ◦ DOWNLOAD, ◦ UPLOAD; • error: A brief description of the error. • debugError; The error's detailed message string.
Notes	Supported since v1.4.0.

3.2 ONJOBCOMPLETED

Signature	<code>void onJobCompleted();</code>
Description	This callback is called when the process of checking for updates is completed without errors.
Parameters	None.
Notes	Supported since v1.4.0.

3.3 ONDOWNLOADSTATUS

Signature	<code>void onDownloadStatus(String applicationName, String fileName, DownloadStatus status);</code>
Description	This function notifies the calling app of the file download status.
Parameters	<ul style="list-style-type: none"> • applicationName: The name of the application. • fileName: The name of the apk file. • status: The following is a list of the available status options: <ul style="list-style-type: none"> ◦ DOWNLOAD_NOT_STARTED, ◦ INVALID_URL, ◦ UNABLE_TO_DOWNLOAD_THE_FILE, ◦ DOWNLOAD_FAILED, ◦ DOWNLOAD_IN_PROGRESS,

	○ SUCCESSFUL_DOWNLOAD;
Notes	Supported since v1.4.0.

3.4 ONDOWNLOADPROGRESS

Signature	<code>void onDownloadProgress(String fileName, int progressPercentage, long fileLength, long currentDownloadLength);</code>
Description	This function notifies the calling app of the file download progress.
Parameters	<ul style="list-style-type: none"> • <code>fileName</code>: The name of the apk file. • <code>progressPercentage</code>: The download percentage. • <code>fileLength</code>: The file size. • <code>currentDownloadLength</code>: The total number of bytes that have been downloaded.
Notes	Supported since v1.4.0.

3.5 ONDOWNLOADJOBNUMBER

Signature	<code>void onDownloadJobNumber(int currentJobNumber, int totalNumberOfDownload);</code>
Description	This function notifies the calling app of the downloading file position in the download queue.
Parameters	<ul style="list-style-type: none"> • <code>currentJobNumber</code>: The file's position in the queue. • <code>totalNumberDownload</code>: The total number of files in the queue.
Notes	Supported since v1.4.0.

3.6 ONFIRMWAREUPGRADESTATUS

Signature	<code>void onFirmwareUpgradeStatus(int ret);</code>
Description	This function returns the status of the firmware upgrade process.
Parameters	<p><code>ret</code>: the returned code</p> <ul style="list-style-type: none"> • 0: Success. • All other values: unsuccessful operation.
Notes	Supported since v1.4.0.

3.7 ONKEYINJECTIONRETURNCODE

Signature	<code>void onKeyInjectionReturnCode(int ret);</code>
Description	This callback returns the value of the TMS key injection when TMS is initialized.
Parameters	ret: the returned code <ul style="list-style-type: none"> • 0: Success. • All other values: unsuccessful operation.
Notes	Supported since v1.4.0.

3.8 ONREQUESTEDFILENOTFOUND

Signature	<code>void onRequestedFileNotFound(TMS_FILE_TYPE tmsFileType);</code>
Description	This callback returns errors when the target application, firmware, parameter or additional file is not found.
Parameters	Type of TMS_FILE_TYPE <ul style="list-style-type: none"> • APPLICATION, • FIRMWARE, • CONFIGURATION_PARAMETERS, • ADDITIONAL_FILES;
Notes	Supported since v1.4.0.

3.9 ONAPPINSTALLCOMPLETED

Signature	<code>void onAppInstallCompleted(String appName, int installResult);</code>
Description	This function notifies the calling app of an app installation completion, along with the result.
Parameters	<ul style="list-style-type: none"> • appName: the installed app name. • installResult: the result of the app installation. <ul style="list-style-type: none"> ◦ 0: Success. ◦ All other values: unsuccessful operation.
Notes	Supported since v1.8.0.

3.10 ONADDITIONALFILESRECEIVED

Signature	<code>void onAdditionalFilesReceived();</code>
Description	This callback is called when an additional file is found, and the file's package name matches that of the calling app.
Parameters	None.
Notes	Supported since v1.4.0.

3.11 ONADDITIONALFILESHASHRECEIVED

Signature	<code>void onAdditionalFilesHashReceived(String hashCode);</code>
Description	This callback returns the hash code of the additional file with the package name that matches that of the calling app.
Parameters	hashCode: The hash code of the additional file.
Notes	Supported since v1.8.0.

3.12 ONCONFIGURATIONPARAMETERSRECEIVED

Signature	<code>void onConfigurationParametersReceived();</code>
Description	This callback is called when the configuration parameter file is found and the file's package name matches that of the calling app.
Parameters	None.
Notes	Supported since v1.4.0.

3.13 ONCONFIGURATIONPARAMETERSHASHRECEIVED

Signature	<code>void onConfigurationParametersHashReceived(String hashCode);</code>
Description	This callback returns the hash code of the configuration parameter file with the package name that matches that of the calling app.
Parameters	hashCode: The hash code of the configuration parameter file.

Notes	Supported since v1.8.0.
--------------	-------------------------

3.14 ONSILENTINSTALLALLLASTDOWNLOADSFINISHED

Signature	<pre>Void onSilentInstallAllLastDownloadsFinished(FirmwareUpdateStatus status, int numberOfApkDownloaded, int numberOfApkInstalled, ArrayList<String> installedAppList);</pre>
Description	This callback returns the result of silentInstallLastAllDownloads command.
Parameters	<ul style="list-style-type: none"> FirmwareUpdateStatus <ul style="list-style-type: none"> FILE_FOUND_AND_START_SUCCESSFUL FILE_FOUND_BUT_START_FAILED FILE_NOT_FOUND numberOfApkDownloaded: The total number of downloaded apk files. numberOfApkInstalled: The total number of installed apk files. installedAppList: A list of all the installed apps' info.
Notes	Supported since v1.8.0

3.15 ONLOCATIONDETECTED

Signature	<pre>void onLocationDetected(String location);</pre>
Description	This callback returns the location info that is currently stored in ATMSAgent. The callback will always be triggered. The parameter will be empty if the location info is not available.
Parameters	location: The location info.
Notes	<p>Supported since v1.8.0.</p> <p>The response location data is not in JSON format</p>

3.16 ONCELLULARDOWNLOADSET

Signature	<code>void onCellularDownloadSet(boolean cellularDownloadSet)</code>
Description	This callback returns the updated boolean value of cellular download switch in ATMSAgent, which was set by <code>setCellularDownload()</code> api.
Parameters	<code>cellularDownloadSet</code> : A boolean value that indicates cellular download is set or not.
Notes	Supported since v1.8.0.

3.17 ONDELETEINTERNALSTORAGEFINISHED

Signature	<code>Void onDeleteInternalStorageFinished(boolean isAllFileDeleted, ArrayList<String> failList)</code>
Description	This callback returns the response after <code>deleteInternalStorage()</code> is called.
Parameters	<ul style="list-style-type: none"> <code>isAllFileDeleted</code>: The flag that indicates if all files are deleted. <code>failList</code>: An array list that contains all file names that failed to be deleted. The list will be empty if all files are deleted successfully.
Notes	Supported since v1.8.0.

3.18 ONSETAPPLICATIONPARAMETERRESPONSE

Signature	<code>Void onSetApplicationParameterResponse(final AMP360Response amp360Response)</code>
Description	This callback returns the server response when api <code>setApplicationParameter()</code> is called and server communication is successful.
Parameters	<ul style="list-style-type: none"> <code>amp360Response</code>: An object of class <code>AMP360Response</code> which contains all server response info.
Notes	Supported since v1.8.4

3.19 ONSETAPPLICATIONPARAMETERFAIL

Signature	<code>Void onSetApplicationParameterFail(final String errorDesc)</code>
Description	This callback returns the error when api <code>setApplicationParameter()</code> is called and server communication fails.
Parameters	<ul style="list-style-type: none"> • <code>errorDesc</code>: The server communication failure error description in text format.
Notes	Supported since v1.8.4

3.20 ONGETAPPLICATIONPARAMETERSRESPONSE

Signature	<code>Void onGetApplicationParametersResponse(final AMP360Response amp360Response)</code>
Description	This callback returns the server response when api <code>getApplicationParameters()</code> is called and server communication is successful.
Parameters	<ul style="list-style-type: none"> • <code>amp360Response</code>: An object of class <code>AMP360Response</code> which contains all server response info.
Notes	Supported since v1.8.4

3.21 ONGETAPPLICATIONPARAMETERSFAIL

Signature	<code>Void onGetApplicationParametersFail(final String errorDesc)</code>
Description	This callback returns the error when api <code>getApplicationParameters()</code> is called and server communication fails.
Parameters	<ul style="list-style-type: none"> • <code>errorDesc</code>: The server communication failure error description in text format.
Notes	Supported since v1.8.4

3.22 (DEPRECATED)ONSETCONFIGURATIONPARAMETERS TATUS

Signature	<code>void onSetConfigurationParameterStatus(int respCode, String respDesc)</code>
Description	This callback returns the server response after setConfigurationParameter() is called.
Parameters	<ul style="list-style-type: none">• respCode: the code indicates the status of the call.• respDesc: the description of the status and code.
Notes	<ul style="list-style-type: none">• Supported since v1.8.0.• All response values are provided by the server via ATMSAgent.• Deprecated since v1.8.4.