

# Android 16 NFC API and static config Changes

Mar, 2025



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# NXP Proprietary API Changes in A16 - Card Emulation Feature

SNO	Exiting NxpNfcExtension API in A15	Proposed New AOSP API in A16	Remarks
1	void DefaultRouteSet(int routeLoc, boolean fullPower, boolean lowPower, boolean noPower);	public void overwriteRoutingTable(int protocol,int technology, int <b>emptyAid</b> ,int systemCode)	1. New AOSP API does not support power state input as argument rather application shall rely on "Secure NFC" feature as alternative. <a href="#">CardEmulationManager.java - Android Code Search</a>
2	void MifareDesfireRouteSet(int routeLoc, boolean fullPower, boolean lowPower, boolean noPower)	public void overwriteRoutingTable(int <b>protocol</b> ,int technology, int emptyAid,int systemCode)	Same as SNO#1
3	void MifareCLTRouteSet(int routeLoc, boolean fullPower, boolean lowPower, boolean noPower);	public void overwriteRoutingTable(int protocol,int <b>technology</b> , int emptyAid,int systemCode)	Same as SNO#1
4	void NfcFRouteSet(int routeLoc, boolean fullPower, boolean lowPower, boolean noPower);	public void overwriteRoutingTable(int protocol,int <b>technology</b> , int emptyAid,int systemCode)	Same as SNO#1



## NXP Proprietary API Changes in A16 – T4T and eSE Feature

SNO	Exiting NxpNfcExtension API in A15	Proposed New AOSP API in A16	Remarks
1	int doWriteT4tData(in byte[] fileId, in byte[] data, int length);	public @WriteDataStatus int writeData(@IntRange(from = 0, to = 65535) int fileId, @NonNull byte[] data)	T4tNdefNfcee.java
2	byte[] doReadT4tData(in byte[] fileId);	public byte[] readData(@IntRange(from = 0, to = 65535) int fileId) {	
3	NA	public NdefCcFileInfo readCcfile()	This API newly added in A16
4	NA	int clearNdefData();	This API newly added in A16
5	NA	boolean isNdefOperationOngoing();	This API newly added in A16
6	NA	boolean isNdefNfceeEmulationSupported();	This API newly added in A16
7	int[] getActiveSecureElementList(String pkg);	List<String> getActiveNfceeList();	
8	void changeDiscoveryTech(IBinder binder, int pollTech, int listenTech);	setDiscoveryTechnology	Deprecated as similar API is present in AOSP



## Un-Supported NXP Proprietary API's in A16

SNO	Exiting NxpNfcExtension API in A15	Proposed New AOSP API in A16	Remarks
1	int getMaxAidRoutingTableSize();	NA	Deprecated. Not used. These API are used because now Listen Mode Routing table is defined in number of bytes and not in number of AID routes.
2	int getCommittedAidRoutingTableSize();	NA	
3	int selectUicc(int uiccSlot);	NA	Deprecated.This API used for static UICC switching. Now, we support dual UICC and both UICC's can be active at same time. Based on the payment application, data can be routed to UICC1 or UICC2.
4	int getSelectedUicc();	NA	
5	int configureSecureReader(boolean on, String readerType);	NA	
6	void enableWlc(INxpWlcCallBack callback);	NA	Deprecated and AOSP WLC recommended
7	void disableWlc(INxpWlcCallBack callback);	NA	Deprecated and AOSP WLC recommended
8	boolean isWlcEnabled();	NA	Deprecated and AOSP WLC recommended
9	void updateStatus(int status)	NA	Deprecated and AOSP WLC recommended



## NXP Proprietary API Changes in A16 – Transit Config Feature

SNO	Exiting NxpNfcExtension API in A15	Proposed NxpNfcExtension API in A16	Remarks
1	int setConfig(String configs, String pkg)	public boolean setConfig(String configs)	Allows to update the configuration options at runtime.

## Un-Supported NXP Proprietary API's in A16

SNO	Exiting NxpNfcExtension API in A15	Proposed New AOSP API in A16	Remarks
1	int activateSelInterface();	boolean setControllerAlwaysOn(ENABLE_EE)	
2	int deactivateSelInterface();	boolean setControllerAlwaysOn(false)	



## NXP Proprietary API Changes in A16 – LxDebug Feature

SNO	Exiting NxpNfcExtension API in A15	Proposed NxpNfcExtension API in A16	Impact
1	com.android.nfc.action.ACTION_EXTENDED_FIELD_TIMEOUT	ILxDebugCallbacks.java public void onEFDMLTimeout()	1. Updated smart_terminal_recognizer_lib to be used. 2. STR should have system application privilege
2	com.android.nfc.action.LX_DATA	ILxDebugCallbacks.java public void onLxDataReceive(byte[] lxDebugData)	



# Un-Supported NXP Proprietary Static Configurations in A16

SNO	Static Configuration	Alternative Proposal	Remarks
1	NXP_CE_ROUTE_STRICT_DISABLE	Not used.	NA
2	DEFAULT_AID_PWR_STATE	Not Supported	Static power state configuration does not supported in Android-16
3	DEFAULT_DESFIRE_PWR_STATE	Not Supported	Same as SNO#2
4	DEFAULT_MIFARE_CLT_PWR_STATE	Not Supported	Same as SNO#2
5	DEFAULT_FELICA_CLT_PWR_STATE	Not Supported	Same as SNO#2
6	DEFAULT_T4TNFCEE_AID_POWER_STATE	Not Supported	Same as SNO#2
7	NXP_CHECK_DEFAULT_PROTO_SE_ID	Not Supported	
8	AID_MATCHING_PLATFORM	Not Used	NA
9	FORWARD_FUNCTIONALITY_ENABLE	Not Supported	Check and update the reference
10	DEFAULT_SYS_CODE_PWR_STATE	Not Supported	Same as SNO#2
11	NXP_CE_PRIORITY_ENABLED	not supported	
12	SCREEN_OFF_POWER_STATE	Not Used	



## New NFC API from AOSP in Android-16

SNO	Feature	New API added in Android-16	Remarks
1	Vendor Extension Support – NfcOemExtension.java	1. RF Discovery management API and overwriteRoutingTable APIs 2. OEMExtensionCallbacks covers Tag dispatch status, Routing and CE events, RF state updates	<a href="#">NfcOemExtension.java - Android Code Search</a>





SECURE CONNECTIONS  
FOR A SMARTER WORLD

