Training API Guides

Reference

Tools

Google Services

Samples

Android APIs API level: 20 \$
android.location
android.media.au
android.media.eff
android.met
android.net
android.net.http
android.net.nsd
android.net.rtp

wifiP2pManager

Summary: Nested Classes | Constants | Methods | Inherited Methods | [Expand All] Added in API level 14

extends Object

java.lang.Object

sandroid.net.wifi.p2p.WifiP2pManager

Class Overview

This class provides the API for managing Wi-Fi peer-to-peer connectivity. This lets an application discover available peers, setup connection to peers and query for the list of peers. When a p2p connection is formed over wifi, the device continues to maintain the uplink connection over mobile or any other available network for internet connectivity on the device.

The API is asynchronous and responses to requests from an application are on listener callbacks provided by the application. The application needs to do an initialization with initialize (Context, Looper, WifiP2pManager.ChannelListener) before doing any p2p operation.

Most application calls need a WifiP2pManager.ActionListener instance for receiving callbacks onSuccess() or onFailure(int). Action callbacks indicate whether the initiation of the action was a success or a failure. Upon failure, the reason of failure can be one of ERROR, P2P_UNSUPPORTED or BUSY.

An application can initiate discovery of peers with discoverPeers (WifiP2pManager.Channel, WifiP2pManager.ActionListener). An initiated discovery request from an application stays active until the device starts connecting to a peer, forms a p2p group or there is an explicit stopPeerDiscovery (WifiP2pManager.Channel, WifiP2pManager.ActionListener). Applications can listen to WIFI_P2P_DISCOVERY_CHANGED_ACTION to know if a peer-to-peer discovery is running or stopped. Additionally, WIFI_P2P_PEERS_CHANGED_ACTION indicates if the peer list has changed.

When an application needs to fetch the current list of peers, it can request the list of peers with requestPeers(WifiP2pManager.Channel, WifiP2pManager.PeerListListener). When the peer list is available onPeersAvailable(WifiP2pDeviceList) is called with the device list.

An application can initiate a connection request to a peer through connect(WifiP2pManager.Channel, WifiP2pConfig, WifiP2pManager.ActionListener). See WifiP2pConfig for details on setting up the configuration. For communication with legacy Wi-Fi devices that do not support p2p, an app can create a group using createGroup(WifiP2pManager.Channel, WifiP2pManager.ActionListener) which creates an access point whose details can be fetched with requestGroupInfo(WifiP2pManager.Channel, WifiP2pManager.GroupInfo(Listener).

After a successful group formation through createGroup(WifiP2pManager.Channel, WifiP2pManager.ActionListener) or through connect(WifiP2pManager.Channel, WifiP2pManager.Channel, WifiP2pManager.Channe

With peer discovery using discoverPeers (WifiP2pManager. Channel, WifiP2pManager. ActionListener), an application discovers the neighboring peers, but has no good way to figure out which peer to establish a connection with. For example, if a game application is interested in finding all the neighboring peers that are also running the same game, it has no way to find out until after the connection is setup. Pre-association service discovery is meant to address this issue of filtering the peers based on the running services.

With pre-association service discovery, an application can advertise a service for a application on a peer device prior to a connection setup between the devices. Currently, DNS based service discovery (Bonjour) and Upnp are the higher layer protocols supported. Get Bonjour resources at dns-sd.org and Upnp resources at upnp.org As an example, a video application can discover a Upnp capable media renderer prior to setting up a Wi-fip2p connection with the device.

An application can advertise a Upnp or a Bonjour service with a call to addLocalService(WifiP2pManager.Channel, WifiP2pServiceInfo, WifiP2pManager.ActionListener). After a local service is added, the framework automatically responds to a peer application discovering the service prior to establishing a p2p connection. A call to removeLocalService(WifiP2pManager.Channel, WifiP2pServiceInfo, WifiP2pManager.ActionListener) removes a local service and clearLocalServices(WifiP2pManager.Channel, WifiP2pManager.ActionListener) can be used to clear all local services.

An application that is looking for peer devices that support certain services can do so with a call to discover Services (WifiP2pManager.Channel, WifiP2pManager.ActionListener). Prior to initiating the discovery, application can add service discovery request with a call to addServiceRequest(WifiP2pManager.Channel, WifiP2pServiceRequest, WifiP2pManager.ActionListener), remove a service discovery request with a call to removeServiceRequest(WifiP2pManager.Channel, WifiP2pServiceRequest, WifiP2pManager.ActionListener) or clear all requests with a call to clearServiceRequests(WifiP2pManager.Channel, WifiP2pManager.ActionListener). When no service requests remain, a previously running service discovery will stop. The application is notified of a result of service discovery request through listener callbacks set through setDnsSdResponseListeners(WifiP2pManager.Channel, WifiP2pManager.DnsSdServiceResponseListener, WifiP2pManager.DnsSdTxtRecordListener) for Bonjour or setUpnpServiceResponseListener(WifiP2pManager.Channel, WifiP2pManager.UpnpServiceResponseListener) for Upnp.

Note: Registering an application handler with initialize (Context, Looper, WifiP2pManager.ChannelListener) requires the permissions ACCESS_WIFI_STATE and CHANGE_WIFI_STATE to perform any further peer-to-peer operations. Get an instance of this class by calling Context.getSystemService(Context.WIFI_P2P_SERVICE).

See Also

WifiP2pConfig

WifiP2pInfo

WifiP2pGroup

WifiP2pDevice

android nfo Interfaces

android.net.sip

android.net.wifi

android.net.wifi.

WifiP2pManager.A WifiP2pManager.C WifiP2pManager.D WifiP2pManager.D WifiP2pManager.D WifiP2pManager.P WifiP2pManager.S WifiP2pManager.S WifiP2pManager.U

Classes

WifiP2pConfig WifiP2pDevice WifiP2pDeviceList WifiP2pGroup WifiP2pInfo WifiP2pManager WifiP2pManager.C

Use Tree Navigation

++

Summary

Nest ed Cla	Nested Classes		
interface	WifiP2pManager.ActionListener	Interface for callback invocation on an application action	
class	WifiP2pManager.Channel	A channel that connects the application to the Wifi p2p framework.	
interface	WifiP2pManager.ChannelListener	Interface for callback invocation when framework channel is lost	
interface	WifiP2pManager.ConnectionInfoListener	Interface for callback invocation when connection info is available	
interface	WifiP2pManager.DnsSdServiceResponseListener	Interface for callback invocation when Bonjour service discovery response is received	
interface	WifiP2pManager.DnsSdTxtRecordListener	Interface for callback invocation when Bonjour TXT record is available for a service	
interface	WifiP2pManager.GroupInfoListener	Interface for callback invocation when group info is available	
interface	WifiP2pManager.PeerListListener	Interface for callback invocation when peer list is available	
interface	WifiP2pManager.ServiceResponseListener	Interface for callback invocation when service discovery response other than Upnp or Bonjour is received	
interface	WifiP2pManager.UpnpServiceResponseListener	Interface for callback invocation when upnp service discovery response is received	

Constants					
int	BUSY	Passed with onFailure(int).			
int	ERROR	Passed with onFailure(int).			
String	EXTRA_DISCOVERY_STATE	The lookup key for an int that indicates whether p2p discovery has started or stopped.			
String	EXTRA_NETWORK_INFO	The lookup key for a NetworkInfo object associated with the p2p network.			
String	EXTRA_P2P_DEVICE_LIST	The lookup key for a WifiP2pDeviceList object representing the new peer list when WIFI_P2P_PEERS_CHANGED_ACTION broadcast is sent.			
String	EXTRA_WIFLP2P_DEVICE	The lookup key for a WifiP2pDevice object Retrieve with getParcelableExtra(String).			
String	EXTRA_WIFL_P2P_GROUP	The lookup key for a WifiP2pGroup object associated with the p2p network.			
String	EXTRA_WIFL_P2P_INFO	The lookup key for a WifiP2pInfo object Retrieve with getParcelableExtra(String).			
String	EXTRA_WIFLSTATE	The lookup key for an int that indicates whether Wi-Fi p2p is enabled or disabled.			
int	NO_SERVICE_REQUESTS	Passed with onFailure(int).			
int	P2P_UNSUPPORT ED	Passed with onFailure(int).			
String	WIFL_P2P_CONNECTION_CHANGED_ACTION	Broadcast intent action indicating that the state of Wi-Fi p2p connectivity has changed.			
String	WIFL_P2P_DISCOVERY_CHANGED_ACTION	Broadcast intent action indicating that peer discovery has either started or stopped.			
int	WIFL_P2P_DISCOVERY_STARTED	p2p discovery has started			
int	WIFL_P2P_DISCOVERY_STOPPED	p2p discovery has stopped			
String	WIFLP2P_PEERS_CHANGED_ACTION	Broadcast intent action indicating that the available peer list has changed.			
String	WIFL_P2P_STATE_CHANGED_ACTION	Broadcast intent action to indicate whether Wi-Fi p2p is enabled or disabled.			
int	WIFL_P2P_STATE_DISABLED	Wi-Fi p2p is disabled.			
int	WIFL_P2P_STATE_ENABLED	Wi-Fi p2p is enabled.			
String	WIFL_P2P_THIS_DEVICE_CHANGED_ACTION	Broadcast intent action indicating that this device details have changed.			

Public Methods	ublic Methods	
void	addLocalService (WifiP2pManager.Channel c, WifiP2pServiceInfo servInfo, WifiP2pManager.ActionListene Register a local service for service discovery.	
void	addServiceRequest (WifiP2pManager.Channel c, WifiP2pServiceRequest req, WifiP2pManager.ActionListe Add a service discovery request.	
void	cancelConnect (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Cancel any ongoing p2p group negotiation The function call immediately returns after sending a connection cancellation request to the framewo	

void	clearLocalServices (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Clear all registered local services of service discovery.
void	clearServiceRequests (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Clear all registered service discovery requests.
void	connect (WifiP2pManager.Channel c, WifiP2pConfig config, WifiP2pManager.ActionListener listener) Start a p2p connection to a device with the specified configuration.
void	createGroup (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Create a p2p group with the current device as the group owner.
void	discoverPeers (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Initiate peer discovery.
void	discoverServices (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Initiate service discovery.
WifiP2pManager.Channel	initialize (Context srcContext, Looper srcLooper, WifiP2pManager.ChannelListener listener) Registers the application with the Wi-Fi framework.
void	removeGroup (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Remove the current p2p group.
void	removeLocalService (WifiP2pManager.Channel c, WifiP2pServiceInfo servInfo, WifiP2pManager.ActionList Remove a registered local service added with addLocalService(WifiP2pManager.Channel, WifiP2pServiceInfo The function call immediately returns after sending a request to remove a local service to the framework
void	removeServiceRequest (WifiP2pManager.Channel c, WifiP2pServiceRequest req, WifiP2pManager.Actionl Remove a specified service discovery request added with addServiceRequest(WifiP2pManager.Channel, Wi The function call immediately returns after sending a request to remove service discovery request to the
void	requestConnectionInfo (WifiP2pManager.Channel c, WifiP2pManager.ConnectionInfoListener listener) Request device connection info.
void	request GroupInfo (Wif iP2pManager.Channel c, Wif iP2pManager.GroupInfoListener listener) Request p2p group info.
void	request Peers (WifiP2pManager.Channel c, WifiP2pManager.PeerListListener listener) Request the current list of peers.
void	setDnsSdResponseListeners (WifiP2pManager.Channel c, WifiP2pManager.DnsSdServiceResponseListeners) Register a callback to be invoked on receiving Bonjour service discovery response.
void	set ServiceResponseListener (WifiP2pManager.Channel c, WifiP2pManager.ServiceResponseListener lister Register a callback to be invoked on receiving service discovery response.
void	setUpnpServiceResponseListener (WifiP2pManager.Channel c, WifiP2pManager.UpnpServiceResponseLis Register a callback to be invoked on receiving upnp service discovery response.
void	stopPeerDiscovery (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Stop an ongoing peer discovery The function call immediately returns after sending a stop request to the framework.

Inherited Methods

[Expand]

► From class java.lang.Object

Constants

public static final int BUSY

Added in API level 14

 $Passed\ with\ on Failure (int).\ Indicates\ that\ the\ operation\ failed\ because\ the\ framework\ is\ busy\ and\ unable\ to\ service\ the\ request$

public static final int ERROR

Constant Value: 2 (0x00000002)

Added in API level 14

Passed with on Failure (int). Indicates that the operation failed due to an internal error.

Constant Value: 0 (0x00000000)

$public\ static\ final\ {\color{blue} String\ EXTRA_DISCOVERY_STATE}$

Added in API level 16

The lookup key for an int that indicates whether p2p discovery has started or stopped. Retrieve it with getIntExtra(String, int).

See Also

WIFI_P2P_DISCOVERY_STARTED WIFI_P2P_DISCOVERY_STOPPED

Constant Value: "discoveryState"

public static final String EXTRA_NET WORK_INFO

Added in API level 14

 $The \ lookup\ key\ for\ a\ Network Info\ object\ associated\ with\ the\ p2p\ network.\ Retrieve\ with\ getParcelable Extra (String).$

Constant Value: "networkInfo"

public static final String EXTRA_P2P_DEVICE_LIST

Added in API level 18

The lookup key for a WifiP2pDeviceList object representing the new peer list when WIFI_P2P_PEERS_CHANGED_ACTION broadcast is sent.

Retrieve with getParcelableExtra(String).

Constant Value: "wifiP2pDeviceList"

public static final String EXTRA_WIFI_P2P_DEVICE

Added in API level 14

The lookup key for a WifiP2pDevice object Retrieve with getParcelableExtra(String).

Constant Value: "wifiP2pDevice"

public static final String EXTRA_WIFI_P2P_GROUP

Added in API level 18

The lookup key for a WifiP2pGroup object associated with the p2p network. Retrieve with getParcelableExtra(String).

Constant Value: "p2pGroupInfo"

public static final String EXTRA_WIFI_P2P_INFO

Added in API level 14

The lookup key for a WifiP2pInfo object Retrieve with getParcelableExtra(String).

Constant Value: "wifiP2pInfo"

public static final String EXTRA_WIFI_STATE

Added in API level 14

The lookup key for an int that indicates whether Wi-Fi p2p is enabled or disabled. Retrieve it with getIntExtra(String, int).

Saa Also

WIFI_P2P_STATE_DISABLED WIFI_P2P_STATE_ENABLED

Constant Value: "wifi_p2p_state"

public static final int NO_SERVICE_REQUESTS

Added in API level 16

Passed with onFailure(int). Indicates that the discoverServices(WifiP2pManager.Channel, WifiP2pManager.ActionListener) failed because no service requests are added. Use addServiceRequest(WifiP2pManager.Channel, WifiP2pServiceRequest, WifiP2pManager.ActionListener) to add a service request.

Constant Value: 3 (0x00000003)

public static final int P2P_UNSUPPORTED

Added in API level 14

Passed with on Failure (int). Indicates that the operation failed because p2p is unsupported on the device.

Constant Value: 1 (0x00000001)

public static final String WIFI_P2P_CONNECTION_CHANGED_ACTION

Added in API level 14

Broadcast intent action indicating that the state of Wi-Fi p2p connectivity has changed. One extra EXTRA_WIFI_P2P_INFO provides the p2p connection info in the form of a WifiP2pInfo object. Another extra EXTRA_NETWORK_INFO provides the network info in the form of a NetworkInfo. A third extra provides the details of the group.

See Also

EXTRA_WIFI_P2P_INFO
EXTRA_NETWORK_INFO
EXTRA_WIFI_P2P_GROUP

Constant Value: "android.net.wifi.p2p.CONNECTION_STATE_CHANGE"

public static final String WIFI_P2P_DISCOVERY_CHANGED_ACTION

Added in API level 16

 $Broadcast\ intent\ action\ indicating\ that\ peer\ discovery\ has\ either\ started\ or\ stopped.\ One\ extra\ {\tt EXTRA_DISCOVERY_STATE}\ indicates\ whether\ discovery\ has\ started\ or\ stopped.$

Note that discovery will be stopped during a connection setup. If the application tries to re-initiate discovery during this time, it can fail.

Constant Value: "android.net.wifi.p2p.DISCOVERY_STATE_CHANGE"

public static final int WIFI_P2P_DISCOVERY_STARTED

Added in API level 16

p2p discovery has started

See Also

WIFI P2P DISCOVERY CHANGED ACTION

Constant Value: 2 (0x00000002)

public static final int WIFI_P2P_DISCOVERY_STOPPED

Added in API level 16

p2p discovery has stopped

See Also

WIFI_P2P_DISCOVERY_CHANGED_ACTION

Constant Value: 1 (0x00000001)

public static final String WIFI_P2P_PEERS_CHANGED_ACTION

Added in API level 14

Broadcast intent action indicating that the available peer list has changed. This can be sent as a result of peers being found, lost or updated.

An extra EXTRA_P2P_DEVICE_LIST provides the full list of current peers. The full list of peers can also be obtained any time with requestPeers (WifiP2pManager. Channel, WifiP2pManager. PeerListListener).

See Also

EXTRA_P2P_DEVICE_LIST

Constant Value: "android.net.wifi.p2p.PEERS_CHANGED"

public static final String WIFI_P2P_STATE_CHANGED_ACTION

Added in API level 14

Broadcast intent action to indicate whether Wi-Fi p2p is enabled or disabled. An extra EXTRA_WIFI_STATE provides the state information as int.

See Also

EXTRA_WIFI_STATE

Constant Value: "android.net.wifi.p2p.STATE_CHANGED"

public static final int WIFI_P2P_STATE_DISABLED

Added in API level 14

Wi-Fi p2p is disabled.

See Also

WIFI_P2P_STATE_CHANGED_ACTION

Constant Value: 1 (0x00000001)

public static final int WIFI_P2P_STATE_ENABLED

Added in API level 14

Wi-Fi p2p is enabled.

See Also

WIFI_P2P_STATE_CHANGED_ACTION

Constant Value: 2 (0x00000002)

public static final String WIFI_P2P_THIS_DEVICE_CHANGED_ACTION

Added in API level 14

Broadcast intent action indicating that this device details have changed.

Constant Value: "android.net.wifi.p2p.THIS_DEVICE_CHANGED"

Public Methods

public void addLocalService (WifiP2pManager.Channel c, WifiP2pServiceInfo servInfo, WifiP2pManager.ActionListener listener)

Added in API level 16

Register a local service for service discovery. If a local service is registered, the framework automatically responds to a service discovery request from a peer.

The function call immediately returns after sending a request to add a local service to the framework. The application is notified of a success or failure to add service through listener callbacks on Success () or on Failure (int).

The service information is set through WifiP2pServiceInfo.

or its subclass calls newInstance(String, String, List) or newInstance(String, String, Map) for a Upnp or Bonjour service respectively

The service information can be cleared with calls to removeLocalService(WifiP2pManager.Channel, WifiP2pServiceInfo, WifiP2pManager.ActionListener) or clearLocalServices(WifiP2pManager.Channel, WifiP2pManager.ActionListener).

Parameters

 $c \hspace{1cm} \text{is the channel created at initialize} (\text{Context, Looper, WifiP2pManager.ChannelListener}) \\$

servInfo is a local service information.

listener for callbacks on success or failure. Can be null.

public void addServiceRequest (WifiP2pManager.Channel c, WifiP2pServiceRequest req, WifiP2pManager.ActionListener listener)

Added in API level 16

Add a service discovery request.

The function call immediately returns after sending a request to add service discovery request to the framework. The application is notified of a success or failure to add service through listener callbacks on Success () or on Failure (int).

After service discovery request is added, you can initiate service discovery by discoverServices (WifiP2pManager.Channel, WifiP2pManager.ActionListener).

The added service requests can be cleared with calls to removeServiceRequest(Channel, WifiP2pServiceRequest, ActionListener) or

clearServiceRequests(Channel, ActionListener)

Parameters

С is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

req is the service discovery request.

for callbacks on success or failure. Can be null. listener

public void cancelConnect (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Added in API level 14

Cancel any ongoing p2p group negotiation

The function call immediately returns after sending a connection cancellation request to the framework. The application is notified of a success or failure to initiate cancellation through listener callbacks on Success() or on Failure (int).

С is the channel created at initialize (Context, Looper, WifiP2pManager, Channel Listener)

for callbacks on success or failure. Can be null.

public void clearLocalServices (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) d in API level 16

Clear all registered local services of service discovery.

The function call immediately returns after sending a request to clear all local services to the framework. The application is notified of a success or failure to add service through listener callbacks on Success() or on Failure (int).

Parameters

С is the channel created at initialize (Context, Looper, WifiP2pManager. Channel Listener)

listener for callbacks on success or failure. Can be null.

public void clearServiceRequests (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener)

Clear all registered service discovery requests.

The function call immediately returns after sending a request to clear all service discovery requests to the framework. The application is notified of a success or failure to add service through listener callbacks on Success() or on Failure (int).

Parameters

С is the channel created at initialize(Context, Looper, WifiP2pManager, Channel Listener)

for callbacks on success or failure. Can be null. listener

public void connect (WifiP2pManager.Channel c, WifiP2pConfig config, WifiP2pManager.ActionListener listener)

Added in API level 14

Added in API level 16

Start a p2p connection to a device with the specified configuration.

The function call immediately returns after sending a connection request to the framework. The application is notified of a success or failure to initiate connect through listener callbacks on Success() or on Failure (int).

Register for WIFI P2P CONNECTION CHANGED ACTION intent to determine when the framework notifies of a change in connectivity

If the current device is not part of a p2p group, a connect request initiates a group negotiation with the peer.

If the current device is part of an existing p2p group or has created a p2p group with createGroup(WifiP2pManager.Channel, WifiP2pManager.ActionListener), an invitation to join the group is sent to the peer device.

is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener) С

options as described in WifiP2pConfig class config listener for callbacks on success or failure. Can be null

public void createGroup (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Added in API level 14

Create a p2p group with the current device as the group owner. This essentially creates an access point that can accept connections from legacy clients as well as other p2p devices.

Note: This function would normally not be used unless the current device needs to form a p2p connection with a legacy client

The function call immediately returns after sending a group creation request to the framework. The application is notified of a success or failure to initiate group creation through listener callbacks on Success() or on Failure (int).

Application can request for the group details with requestGroupInfo(WifiP2pManager.Channel, WifiP2pManager.GroupInfoListener).

Parameters

is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener) C

listener for callbacks on success or failure. Can be null.

public void discoverPeers (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Added in API level 14

Initiate peer discovery. A discovery process involves scanning for available Wi-Fi peers for the purpose of establishing a connection

The function call immediately returns after sending a discovery request to the framework. The application is notified of a $success\ or\ failure\ to\ initiate\ discovery\ through\ listener\ callbacks\ on Success()\ or\ on Failure(int).$

The discovery remains active until a connection is initiated or a p2p group is formed. Register for WIFI_P2P_PEERS_CHANGED_ACTION intent to determine when the framework notifies of a change as peers are discovered.

Upon receiving a WIFI_P2P_PEERS_CHANGED_ACTION intent, an application can request for the list of peers using requestPeers(WifiP2pManager.Channel, WifiP2pManager.PeerListListener).

Parameters

c is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

listener for callbacks on success or failure. Can be null.

public void discoverServices (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) ded in API level 16

Initiate service discovery. A discovery process involves scanning for requested services for the purpose of establishing a connection to a peer that supports an available service.

The function call immediately returns after sending a request to start service discovery to the framework. The application is notified of a success or failure to initiate discovery through listener callbacks on Success() or on Failure (int).

The services to be discovered are specified with calls to addServiceRequest(WifiP2pManager.Channel, WifiP2pServiceRequest, WifiP2pManager.ActionListener).

The application is notified of the response against the service discovery request through listener callbacks registered by setServiceResponseListener(WifiP2pManager.Channel, WifiP2pManager.ServiceResponseListener) or setDnsSdResponseListeners(WifiP2pManager.Channel, WifiP2pManager.DnsSdServiceResponseListener, WifiP2pManager.DnsSdTxtRecordListener), or setUpnpServiceResponseListener(WifiP2pManager.Channel, WifiP2pManager.UpnpServiceResponseListener).

Parameters

c is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener) listener for callbacks on success or failure. Can be null.

public WifiP2pManager.Channel initialize (Context srcContext, Looper srcLooper, WifiP2pManager.ChannelListener listener)

Added in API level 14

Registers the application with the Wi-Fi framework. This function must be the first to be called before any p2p operations are performed.

Parameters

srcContext is the context of the source

srcLooper is the Looper on which the callbacks are receivied

listener for callback at loss of framework communication. Can be null.

Returns

Channel instance that is necessary for performing any further p2p operations

public void removeGroup (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) Added in API level 14

Remove the current p2p group.

The function call immediately returns after sending a group removal request to the framework. The application is notified of a success or failure to initiate group removal through listener callbacks on Success() or on Failure (int).

Parameters

is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

listener for callbacks on success or failure. Can be null.

public void removeLocalService (WifiP2pManager.Channel c, WifiP2pServiceInfo servInfo, WifiP2pManager.ActionListener listener)

Added in API level 16

Remove a registered local service added with addLocalService(WifiP2pManager.Channel, WifiP2pServiceInfo, WifiP2pManager.ActionListener)

The function call immediately returns after sending a request to remove a local service to the framework. The application is notified of a success or failure to add service through listener callbacks on Success() or on Failure (int).

Parameters

 $c \hspace{1cm} \text{is the channel created at } \text{initialize} (\text{Context, Looper, WifiP2pManager.ChannelListener}) \\$

servInfo is the local service information.

listener for callbacks on success or failure. Can be null.

public void removeServiceRequest (WifiP2pManager.Channel c, WifiP2pServiceRequest req, WifiP2pManager.ActionListener listener)

Added in API level 16

Remove a specified service discovery request added with addServiceRequest(WifiP2pManager.Channel, WifiP2pServiceRequest, WifiP2pManager.ActionListener)

The function call immediately returns after sending a request to remove service discovery request to the framework. The application is notified of a success or failure to add service through listener callbacks on Success() or on Failure (int).

Parameters

c is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

req is the service discovery request.

listener for callbacks on success or failure. Can be null.

Added in API level 14

Request device connection info.

Parameters

is the channel created at initialize(Context, Looper, WifiP2pManager, ChannelListener)

listener for callback when connection info is available. Can be null.

public void requestGroupInfo (WifiP2pManager.Channel c, WifiP2pManager.GroupInfoListener listener)

Added in API level 14

Request p2p group info.

Parameters

c is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

listener for callback when group info is available. Can be null.

public void requestPeers (WifiP2pManager.Channel c, WifiP2pManager.PeerListListener listener) Added in API level 14

Request the current list of peers.

Parameters

c is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

listener for callback when peer list is available. Can be null.

public void setDnsSdResponseListeners (WifiP2pManager.Channel c,

WifiP2pManager.DnsSdServiceResponseListener servListener,

WifiP2pManager.DnsSdTxtRecordListener txtListener)

Added in API level 16

Register a callback to be invoked on receiving Bonjour service discovery response.

see discoverServices(WifiP2pManager.Channel, WifiP2pManager.ActionListener) for the detail.

Parameters

 servListener
 is for listening to a Bonjour service response

 txtListener
 is for listening to a Bonjour TXT record response

$public\ void\ set Service Response Listener\ (Wifi P2p Manager. Channel\ c,$

WifiP2pManager.ServiceResponseListener listener)

Added in API level 16

Register a callback to be invoked on receiving service discovery response. Used only for vendor specific protocol right now. For Bonjour or Upnp, use setDnsSdResponseListeners (WifiP2pManager. Channel, WifiP2pManager. DnsSdServiceResponseListener, WifiP2pManager. DnsSdTxtRecordListener) or setUpnpServiceResponseListener (WifiP2pManager. Channel, WifiP2pManager. UpnpServiceResponseListener) respectively.

see discoverServices(WifiP2pManager.Channel, WifiP2pManager.ActionListener) for the detail.

Parameters

is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

listener for callbacks on receiving service discovery response.

$public\ void\ set Upnp Service Response Listener\ (Wifi P2pManager. Channel\ c,$

WifiP2pManager.UpnpServiceResponseListener listener)

Added in API level 16

Register a callback to be invoked on receiving upnp service discovery response.

 $see \ {\it discoverServices} (WifiP2pManager. Channel, \ WifiP2pManager. Action Listener) \ for the \ detail.$

Parameters

c is the channel created at initialize(Context, Looper, WifiP2pManager.ChannelListener)

listener for callbacks on receiving service discovery response.

public void stopPeerDiscovery (WifiP2pManager.Channel c, WifiP2pManager.ActionListener listener) d in API level 16

Stop an ongoing peer discovery

The function call immediately returns after sending a stop request to the framework. The application is notified of a success or failure to initiate stop through listener callbacks on Success () or on Failure (int).

Parameters

 $c \hspace{1cm} \text{is the channel created at initialize} (\text{Context, Looper, WifiP2pManager.ChannelListener}) \\$

listener for callbacks on success or failure. Can be null.

Except as noted, this content is licensed under Apache 2.0. For details and restrictions, see the Content License. Android $4.4\,\mathrm{r1} - 22\,\mathrm{Sep}\,2014\,22:49$

About Android | Legal | Support