‹ Back to Knox SDK API References page

SAMSUNG Knox SDK API reference

Map API level to SDK version ☐ Filter by API level: 1

public class

VpnPolicy

extends Object

java.lang.Object
↳com.samsung.android.knox.net.vpn.VpnPolicy

Class Overview

This class provides APIs to configure Android VPN settings, including creating, updating, and deleting VPN profiles.

Since

API level 2
MDM 2.0

Summary

Public Methods

boolean	allowOnlySecureConnections(boolean enable) *Deprecated from API level 30 on, please use `allowOnlySecureConnections(boolean)` to allow only secure connections.*
boolean	allowUserAddProfiles(boolean allow) API to prevent the user from creating new VPN profiles.
boolean	allowUserChangeProfiles(boolean allow) API to prevent the user from changing or deleting existent VPN profiles.
boolean	allowUserSetAlwaysOn(boolean allow) *Deprecated from API level 30 on, please use `allowUserSetAlwaysOn(boolean)` to prevent the user from changing Always ON configuration.*
boolean	createProfile(VpnAdminProfile profile) API to create a new VPN profile.
void	deleteProfile(String profileName) *Deprecated from API level 30 on, please use `removeVpnProfile(String)` to delete a VPN profile created by the end-user in Settings.*
String	getAlwaysOnProfile() API to get the VPN profile configured to work in Always ON mode.
List<String>	getDnsDomains(String profileName) API to get the VPN profile DNS search domains.
List<String>	getDnsServers(String profileName) API to get the VPN profile DNS server IP addresses.
List<String>	getForwardRoutes(String profileName) API to get the VPN profile forward routes.
String	getIPSecCaCertificate(String profileName) API to get the CA certificate of a VPN profile.

String	getIPSecPreSharedKey(String profileName) API to get the pre-shared key for a VPN profile.
String	getIPSecUserCertificate(String profileName) API to get the user certificate of a VPN profile.
String	getId(String profileName) API to get the Id of the VPN profile.
String	getIpSecIdentifier(String profileName) API to get the IP security (IPsec) identifier.
String	getL2TPSecret(String profileName) API to get the secret set for an L2TP VPN profile.
String	getOcspServerUrl(String profileName) API to get the OCSP server URL.
String	getServerName(String profileName) API to get the server name of the VPN profile.
String	getState(String profileName) API to get the connection state of the VPN profile.
List<String>	getSupportedConnectionTypes() API to get support VPN Types.
String	getType(String profileName) API to get the VPN profile type.
String	getUserName(String profileName) API to get the user name of the VPN profile.
String	getUserPassword(String profileName) API to get the password of the VPN profile.
String[]	getVpnList() *Deprecated from API level 30 on, please use `getAllVpnProfiles()` to list all VPN connections on the device.*
boolean	isAdminProfile(String profileName)

API to check whether a VPN profile was created by a particular administrator.

boolean	isOnlySecureConnectionsAllowed () *Deprecated from API level 30 on, please use `isOnlySecureConnectionsAllowed()` to check whether only secure connections are allowed.*
boolean	isPPTPEncryptionEnabled (String profileName) API to check if encryption is enabled or disabled for a VPN PPTP profile.
boolean	isUserAddProfilesAllowed () API to verify whether the user is allowed to add new VPN profiles or not.
boolean	isUserChangeProfilesAllowed () API to verify whether the user is allowed to change or delete VPN profiles.
boolean	isUserSetAlwaysOnAllowed () *Deprecated from API level 30 on, please use `isUserSetAlwaysOnAllowed()` to check if the user is allowed to change VPN Always ON mode configuration or not.*
boolean	setAlwaysOnProfile (String profileName) API to set VPN Always ON mode for a certain VPN profile.
boolean	setDnsDomains (String profileName, List<String> searchDomains) API to set the VPN profile DNS search domains.
boolean	setDnsServers (String profileName, List<String> dnsServers) API to set the VPN profile DNS server IP addresses.
boolean	setForwardRoutes (String profileName, List<String> routes) API to set the VPN profile forward routes.
boolean	setIPSecCaCertificate (String profileName, String certificate) API to set the CA certificate for a VPN profile.
boolean	setIPSecPreSharedKey (String profileName, String psk) API to set the pre-shared key for a VPN profile.
boolean	setIPSecUserCertificate (String profileName, String certificate) API to set the user certificate for a VPN profile.

void	setId(String profileName, String id) API to set the Id for a VPN profile.
boolean	setIpSecIdentifier(String profileName, String ipSecIdentifier) API to set the IP security (IPsec) identifier.
boolean	setL2TPSecret(String profileName, boolean enabled, String secret) API to set the secret for an L2TP VPN profile.
boolean	setOcspServerUrl(String profileName, String ocspServerUrl) API to set the OCSP server URL.
boolean	setPPTPEncryptionEnabled(String profileName, boolean enabled) API to enable or disable encryption for a VPN PPTP profile.
void	setProfileName(String oldProfileName, String newProfileName) API to change the profile name of a VPN profile.
void	setServerName(String profileName, String serverName) API to set the VPN server name.
boolean	setUserName(String profileName, String userName) API to set the VPN user name.
boolean	setUserPassword(String profileName, String userPassword) API to set the VPN user password.

Inherited Methods

- From class java.lang.Object

Public Methods

public boolean **allowOnlySecureConnections** (boolean enable) Since: API level 6

> Deprecated from API level 30 on, please use `allowOnlySecureConnections(boolean)` to allow only secure connections.

API to allow only IPsec or SSL/TLS VPN connections.

Returns

`true` if setting allow/disallow only secure connections was successful, else `false`.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to force the Android system VPN client to use either IPSec or SSL/TLS when connecting to networks.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.allowOnlySecureConnections(true);
    if (result) {
        // Policy successfully set. Only VPN connections using
        // IPSec or SSL/TLS are allowed.
    } else {
        //Policy not successfully set
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 6

MDM 4.0

Multiuser Environment

Global Scope

See Also

`isOnlySecureConnectionsAllowed()`

public boolean allowUserAddProfiles (boolean allow)

Since: API level 11

API to prevent the user from creating new VPN profiles.

Parameters

allow `true` to permit VPN profiles creation by users, `false` to block it.

Returns

`true` if the policy was successfully applied, `false` otherwise

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to block VPN profile creation by users. In this case, the menu item to add profiles is disabled.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.allowUserAddProfiles(false);
    if (result) {
        // Policy was applied successfully and profile creation
        // by users is blocked
    } else {
        // Some error occured and the policy was not applied properly
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`isUserAddProfilesAllowed()`

public boolean allowUserChangeProfiles (boolean allow)

Since: API level 11

API to prevent the user from changing or deleting existent VPN profiles.

Parameters

allow `true` to permit user changes (edition and deletion) on VPN profiles, `false` to block it.

Returns

`true` if the policy was successfully applied, `false` otherwise

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to block user changes on existing VPN profiles. In this case, the user is not able to edit or delete profiles. Username and password are not covered by this policy and can always be changed by users.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.allowUserChangeProfiles(false);
    if (result) {
        // Policy was applied successfully and
        // user changes on profiles are blocked
    } else {
        // Some error occured and the policy was not applied properly
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`isUserChangeProfilesAllowed()`

public boolean allowUserSetAlwaysOn (boolean allow)

Since: API level 11

> Deprecated from API level 30 on, please use `allowUserSetAlwaysOn(boolean)` to prevent the user from changing Always ON configuration.

API to prevent the user from changing Always ON configuration.

Parameters

allow `true` to permit user changes on VPN Always ON configuration, `false` to block it.

Returns

`true` if the policy was successfully applied, `false` otherwise

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to block user changes on VPN Always ON mode. The menu item for that configuration is disabled.

NOTE: This API depends on VPN Always ON feature and it is not properly working on all Android devices. Make sure that this feature works on your device before using this API.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.allowUserSetAlwaysOn(false);
    if (result) {
        // Policy was applied successfully and user changes
        // on Always ON settings are blocked
    } else {
        // Some error occured and the policy was not applied properly
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`isUserSetAlwaysOnAllowed()`

public boolean createProfile (VpnAdminProfile profile)

Since: API level 2

API to create a new VPN profile.

Parameters

profile The VPN profile to be created.

Returns

`true` if creation is successful, `false` on failure.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to create a new VPN profile to be provisioned on the device without user interaction. Use the class VpnAdminProfile to complete the VPN profile details and use this API to create the profile.
Notice that, until Knox API 29, the KeyStore must be unlocked for all profile types. See unlockCredentialStorage(String)

NOTE 1: API does not support creating VPN accounts with duplicated name.
NOTE 2: When CC mode is enabled (see setCCMode(boolean)) the following VPN types will not be allowed: VPN_TYPE_PPTP, VPN_TYPE_L2TP_IPSEC_PSK, VPN_TYPE_L2TP_IPSEC_CRT and VPN_TYPE_IPSEC_HYBRID_RSA.
NOTE 3: These connection types aren't supported from API level 35: VPN_TYPE_PPTP, VPN_TYPE_L2TP_IPSEC_PSK, VPN_TYPE_L2TP_IPSEC_CRT, VPN_TYPE_IPSEC_HYBRID_RSA, VPN_TYPE_IPSEC_XAUTH_PSK and VPN_TYPE_IPSEC_XAUTH_RSA.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    // Creating PPTP profile
    VpnAdminProfile profile = new VpnAdminProfile();
    profile.profileName = "PPTP Profile";
    profile.serverName = "127.0.0.1";
    profile.vpnType = VpnAdminProfile.VPN_TYPE_PPTP;
    profile.userName = "username";
    profile.userPassword = "password";
    profile.PPTPEncryptionEnable = true;
    boolean success = vpnPolicy.createProfile(profile);
    if (success) {
        Log.d("VPNPolicy", "PPTP Profile created.");
    } else {
        Log.d("VPNPolicy", "FAILED - PPTP Profile not created.");
    }

    // Creating L2TP/PSK profile
    // Keystore must be unlocked to create this kind of profile
    VpnAdminProfile profile2 = new VpnAdminProfile();
    profile2.profileName = "L2TP PSK Profile";
    profile2.serverName = "127.0.0.1";
    profile2.vpnType = VpnAdminProfile.VPN_TYPE_L2TP_IPSEC_PSK;
    profile2.userName = "username";
    profile2.userPassword = "password";
    profile2.L2TPSecret = "L2tpSecret";
    profile2.IPSecPreSharedKey = "IpsecPreSharedKey";

    success = vpnPolicy.createProfile(profile2);
    if (success) {
        Log.d("VPNPolicy", "L2TP/IPsec PSK profile created!");
    } else {
        Log.d("VPNPolicy", "FAILED - L2TP/IPsec PSK profile not created.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Dependency

name and server name must be set before creating the profile

Multiuser Environment

Global Scope

See Also

`unlockCredentialStorage(String)`

public void deleteProfile (String profileName)

Since: API level 2

> Deprecated from API level 30 on, please use `removeVpnProfile(String)` to delete a VPN profile created by the end-user in Settings.

API to delete an existing VPN profile.

Parameters

profileName The profile name of the VPN profile to be deleted.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to delete a VPN profile provisioned on the device without user interaction.
Notice that, until Knox API 29, the KeyStore must be unlocked for all profile types. See `unlockCredentialStorage(String)`

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Dependency

profile is deleted only if it exists

Multiuser Environment

Global Scope

See Also

`createProfile(VpnAdminProfile)`
`unlockCredentialStorage(String)`

public String getAlwaysOnProfile ()

Since: API level 11

API to get the VPN profile configured to work in Always ON mode.

Returns

the profile name of the VPN connection configured to work in Always ON mode, or `null` if Always ON mode is disabled.

Usage

An administrator can use this API to get the VPN profile that was set to work in Always ON mode. In case that VPN connection stops working, no network traffic is permitted until that connection is restablished.

NOTE: This API depends on VPN Always ON feature and it is not properly working on all Android devices. Make sure that this feature works on your device before using this API.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String result = vpnPolicy.getAlwaysOnProfile();
    if (result != null) {
        Log.d("VPNPolicy","Always ON profile: " + result);
    } else {
        Log.d("VPNPolicy","No profile is set as Always ON or some error occured.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`setAlwaysOnProfile(String)`

public List<String> getDnsDomains (String profileName)

Since: API level 5

API to get the VPN profile DNS search domains.

Parameters

profileName The current profile name of the VPN profile.

Returns

The DNS search domains used for the VPN profile. Returns `null` on failure.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the DNS search domains in the VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    List listDnsDomain = vpnPolicy.getDnsDomains(profileName);
    String outDnsDomain = "";
    for (String dnsDomain : listDnsDomain) {
        outDnsDomain += dnsDomain + " ";
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`setDnsDomains(String, List)`
`unlockCredentialStorage(String)`

public List<String> getDnsServers (String profileName)

Since: API level 5

API to get the VPN profile DNS server IP addresses.

Parameters

profileName The current profile name of the VPN profile.

Returns

The DNS server IP addresses used in the VPN profile. `Null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the DNS server IP addresses in the VPN profile provisioned on the device. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    List listDnsServers = vpnPolicy.getDnsServers(profileName);
    String outDnsServers = "";
    for (String dnsServer : listDnsServers) {
        outDnsServers += dnsServer + " ";
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`setDnsServers(String, List)`
`unlockCredentialStorage(String)`

public List<String> getForwardRoutes (String profileName)

Since: API level 5

API to get the VPN profile forward routes.

Parameters

profileName The current profile name of the VPN profile.

Returns

The forward route addresses in CIDR format. Returns `null` on failure.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to determine the forward routes in the VPN profile provisioned on the device without any user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    List listRoutes = vpnPolicy.getForwardRoutes(profileName);
    String outRoutes = "";
    for (String routes : listRoutes) {
        outRoutes += routes + " ";
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`setForwardRoutes(String, List)`
`unlockCredentialStorage(String)`

public String getIPSecCaCertificate (String profileName)

Since: API level 2

API to get the CA certificate of a VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

The name of the CA certificate for the VPN profile. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the CA certificate of an L2TP/IPsec CRT VPN profile provisioned on the device without user interaction. To properly use this API, an L2TP/IPsec CRT profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileCaCertificate = vpnPolicy.getIPSecCaCertificate(profileName);
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

`setIPSecCaCertificate(String, String)`
`unlockCredentialStorage(String)`

public String getIPSecPreSharedKey (String profileName)

Since: API level 2

API to get the pre-shared key for a VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

The pre-shared key for the VPN profile matching the profile name. `Null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the pre-shared key of an L2TP/IPsec PSK VPN profile provisioned on the device without user interaction. To properly use this API, an L2TP/IPsec PSK profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Dependency

the profile created must be of type L2TP/PSK

Multiuser Environment

Global Scope

See Also

setIPSecPreSharedKey(String, String)
unlockCredentialStorage(String)

public String getIPSecUserCertificate (String profileName)

Since: API level 2

API to get the user certificate of a VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

The name of the user certificate for the VPN profile. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the user certificate of an L2TP/IPsec CRT VPN profile provisioned on the device without user interaction. To properly use this API, an L2TP/IPsec CRT profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See unlockCredentialStorage(String).

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

setIPSecUserCertificate(String, String)
unlockCredentialStorage(String)

public String getId (String profileName)

Since: API level 2

API to get the Id of the VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

The ID of the VPN profile. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the ID of the VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

`unlockCredentialStorage(String)`

public String getIpSecIdentifier (String profileName)

Since: API level 5

API to get the IP security (IPsec) identifier.

Parameters

profileName The current profile name of the VPN profile.

Returns

The IP security identifier used for the VPN L2TP/PSK and XAUTH PSK profile. Returns `null` on failure.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the IP security (IPsec) identifier in the VPN profile provisioned on the device without any user interaction. To properly use this API, a VPN profile must already exist.
Notice that KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    String ipSecIdentifier = vpnPolicy.getIpSecIdentifier(profileName);
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`setIpSecIdentifier(String, String)`

public String **getL2TPSecret** (String profileName)

Since: API level 2

API to get the secret set for an L2TP VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

A string with the L2TP secret. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the secret for an L2TP VPN profile provisioned on the device. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

`setL2TPSecret(String, boolean, String)`
`unlockCredentialStorage(String)`

public String **getOcspServerUrl** (String profileName)

Since: API level 11

API to get the OCSP server URL.

Parameters

profileName The current profile name of the VPN profile.

Returns

The OCSP server URL. `null` if the profile doesn't exist.

Throws

SecurityException If caller does not have required permissions

Usage

Administrator can get Online Certificate Status Protocol (OCSP) server URL. If returned string is empty (zero length) OCSP protocol is disabled.
OCSP is used only with IKEv2 RSA connection type. With other connection types this value is ignored.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String ocspServerUrl = vpnPolicy.getOcspServerUrl(profileName);
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`setOcspServerUrl(String, String)`

public String **getServerName** (String profileName)

Since: API level 2

API to get the server name of the VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

The server name of the VPN profile. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the server name of the VPN profile provisioned on the device without user interaction.
To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Multiuser Environment

Global Scope

See Also

setServerName(String, String)
unlockCredentialStorage(String)

public String getState (String profileName)

Since: API level 2

API to get the connection state of the VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

String with the VPN profile status. Possible return values are CONNECTING, DISCONNECTING, CANCELLED, CONNECTED, IDLE, UNUSABLE, or UNKNOWN. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the connection state of the VPN profile provisioned on the device without any user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See unlockCredentialStorage(String).

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileState = vpnPolicy.getState(profileName);
    Log.w("VPNPolicy", profileName + " state : " + profileState);
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
// Sample Output
// ProfileName state: IDLE
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

unlockCredentialStorage(String)

public List<String> getSupportedConnectionTypes ()

Since: API level 11

API to get support VPN Types.

Returns

List of supported connection types.

Throws

SecurityException If caller does not have required permissions

Usage

Administrator can get a list of supported VPN types. VPN_TYPE_IPSEC_IKEV2_PSK and VPN_TYPE_IPSEC_IKEV2_RSA are only supported by some device models which include strongSwan VPN client. This method can be used to discover if these connection types supported or not.
Possible result list item values are:

VpnAdminProfile.VPN_TYPE_PPTP,

VpnAdminProfile.VPN_TYPE_L2TP_IPSEC_PSK,

VpnAdminProfile.VPN_TYPE_L2TP_IPSEC_CRT,

VpnAdminProfile.VPN_TYPE_IPSEC_HYBRID_RSA,

VpnAdminProfile.VPN_TYPE_IPSEC_XAUTH_PSK,

VpnAdminProfile.VPN_TYPE_IPSEC_XAUTH_RSA,

VpnAdminProfile.VPN_TYPE_IPSEC_IKEV2_PSK and

VpnAdminProfile.VPN_TYPE_IPSEC_IKEV2_RSA

NOTE: These connection types aren't supported from API level 35: VPN_TYPE_PPTP, VPN_TYPE_L2TP_IPSEC_PSK, VPN_TYPE_L2TP_IPSEC_CRT, VPN_TYPE_IPSEC_HYBRID_RSA, VPN_TYPE_IPSEC_XAUTH_PSK and VPN_TYPE_IPSEC_XAUTH_RSA.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    List connectionTypes = vpnPolicy.getSupportedConnectionTypes();
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

VPN_TYPE_PPTP

VPN_TYPE_L2TP_IPSEC_PSK

VPN_TYPE_L2TP_IPSEC_CRT
VPN_TYPE_IPSEC_HYBRID_RSA
VPN_TYPE_IPSEC_XAUTH_PSK
VPN_TYPE_IPSEC_XAUTH_RSA
VPN_TYPE_IPSEC_IKEV2_PSK
VPN_TYPE_IPSEC_IKEV2_RSA
setOcspServerUrl(String, String)

public String getType (String profileName)

Since: API level 2

API to get the VPN profile type.

Parameters

profileName The current profile name of the VPN profile.

Returns

The VPN profile type, `null` if no VPN profile is set.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the type of VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See unlockCredentialStorage(String).

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    // Creating L2TP / PSK profile
    // Keystore must be unlocked to create this kind of profile
    VpnAdminProfile profile = new VpnAdminProfile();
    profile.profileName = "L2TP PSK Profile";
    profile.serverName = "127.0.0.1";
    profile.vpnType = VpnAdminProfile.VPN_TYPE_L2TP_IPSEC_PSK;
    profile.userName = "username";
    profile.userPassword = "password";
    profile.L2TPSecret = "L2tpSecret";
    profile.IPSecPreSharedKey = "IpsecPreSharedKey";

    boolean success = vpnPolicy.createProfile(profile);
    if (success) {
        Log.d("VPNPolicy", "L2TP/IPsec PSK profile created!");
    } else {
        Log.d("VPNPolicy", "FAILED - L2TP/IPsec PSK profile not created.");
    }

    String profileName = "L2TP PSK Profile";
    String profileType = vpnPolicy.getType(profileName);
    Log.d("VPNPolicy", "Profile type : " + profileType);
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}

// Sample Output
// Profile type: L2TP_IPSEC_PSK
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

unlockCredentialStorage(String)

public String getUserName (String profileName)

Since: API level 2

API to get the user name of the VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

String with the user name of the VPN profile. `Null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the user name of the VPN profile provisioned on thedevice without user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileUserName = vpnPolicy.getUserName(profileName);
    Log.w("VPNPolicy", profileName + " username : " + profileUserName);
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

`setUserName(String, String)`
`unlockCredentialStorage(String)`

public String **getUserPassword** (String profileName)

Since: API level 2

API to get the password of the VPN profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

The password of the VPN profile. `null` if the profile does not exist.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the password of the VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

`setUserPassword(String, String)`
`unlockCredentialStorage(String)`

public String[] getVpnList ()

Since: API level 2

> Deprecated from API level 30 on, please use `getAllVpnProfiles()` to list all VPN connections on the device.

API to retrieve a list of VPN profiles on the device.

Returns

A string vector containing the VPN profiles. `Null` if no profile exists.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can get the list of VPNs provisioned on the device that are controlled by a particular administrator.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String[] profileList = vpnPolicy.getVpnList();
    for (String profile : profileList) {
        Log.w("VPNPolicy", profile);
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
// Sample Output
// profile1
// profile2
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Multiuser Environment

Global Scope

See Also

`unlockCredentialStorage(String)`

public boolean isAdminProfile (String profileName)

Since: API level 2

API to check whether a VPN profile was created by a particular administrator.

Parameters

profileName The current profile name of the VPN profile.

Returns

`true` if it is a particular administrator VPN profile, `false` if it is not a particular administrator profile.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to determine whether a particular administrator created a VPN profile.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Multiuser Environment

Global Scope

See Also

`unlockCredentialStorage(String)`

public boolean isOnlySecureConnectionsAllowed ()

Since: API level 6

> Deprecated from API level 30 on, please use `isOnlySecureConnectionsAllowed()` to check whether only secure connections are allowed.

API to check whether only IPSec or SSL/TLS VPN connections are allowed.

Returns

`true` if only IPSec or SSL/TLS VPN connections are allowed, `false` otherwise

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to check if Android system VPN must use either IPSec or SSL/TLS when connecting to networks.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.isOnlySecureConnectionsAllowed();
    if (result) {
        // Only IPSec or SSL/TLS VPN connections are allowed.
    } else {
        //Any VPN connection is allowed.
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 6

MDM 4.0

Multiuser Environment

Global Scope

See Also

`allowOnlySecureConnections(boolean)`

public boolean **isPPTPEncryptionEnabled** (String profileName)

Since: API level 2

API to check if encryption is enabled or disabled for a VPN PPTP profile.

Parameters

profileName The current profile name of the VPN profile.

Returns

`true` if PPTP encryption is enabled, `false` if PPTP encryption is disabled.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to get the encryption status of a PPTP VPN profile provisioned on the device without user interaction. To properly use this API, a PPTP VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

setPPTPEncryptionEnabled(String, boolean)
unlockCredentialStorage(String)

public boolean isUserAddProfilesAllowed ()

Since: API level 11

API to verify whether the user is allowed to add new VPN profiles or not.

Returns

`true` if users are allowed to add new VPN profiles, `false` otherwise

Usage

An administrator can use this API to check if VPN profile creation by users is allowed or not.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.isUserAddProfilesAllowed();
    if (result) {
        // Users are allowed to create VPN profiles
    } else {
        // Users are not permitted to add VPN profiles.
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Since

API level 11
MDM 5.0

Multiuser Environment

Global Scope

See Also

allowUserAddProfiles(boolean)

public boolean isUserChangeProfilesAllowed ()

Since: API level 11

API to verify whether the user is allowed to change or delete VPN profiles.

Returns

`true` if users are allowed to modify or delete VPN profiles, `false` otherwise

Usage

An administrator can use this API to check if user changes on VPN profiles are allowed or not. If not, the user is prevented from editing and deleting VPN profiles. Username and password are not covered by this policy and can always be changed by users.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.isUserChangeProfilesAllowed();
    if (result) {
        // Users are allowed to change VPN profiles.
    } else {
        // Users are not permitted to change VPN profiles.
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`allowUserChangeProfiles(boolean)`

public boolean isUserSetAlwaysOnAllowed ()

Since: API level 11

> Deprecated from API level 30 on, please use `isUserSetAlwaysOnAllowed()` to check if the user is allowed to change VPN Always ON mode configuration or not.

API to verify whether the user is allowed to change VPN Always ON mode configuration or not.

Returns

`true` if users are allowed to modify Always ON settings, `false` otherwise

Usage

An administrator can use this API to check if user changes on VPN Always ON mode are allowed or not.

NOTE: This API depends on VPN Always ON feature and it is not properly working on all Android devices. Make sure that this feature works on your device before using this API.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.isUserSetAlwaysOnAllowed();
    if (result) {
        // Users are allowed to change Always ON configuration.
    } else {
        // Users are not permitted to change Always ON mode.
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Since

API level 11
MDM 5.0

Multiuser Environment

Global Scope

See Also

`allowUserSetAlwaysOn(boolean)`

public boolean setAlwaysOnProfile (String profileName)

Since: API level 11

API to set VPN Always ON mode for a certain VPN profile.

Parameters

profileName the name of the VPN profile to be set as Always ON. If `null`, Always ON mode will be disabled.

Returns

`true` if Always ON mode was successfully set for the given profile, `false` otherwise.

Throws

SecurityException If caller does not have required permissions

Usage

An administrator can use this API to set a VPN profile to work in Always ON mode. In case the VPN connection stops working, no network traffic is permitted until that connection is restablished.

NOTE: The Always ON feature has some mandatory requirements regarding the VPN profile in use:

- it must be previously created;
- it must have an authentication method different from PPTP;
- its Server and DNS server addresses must be provided in IPv4 format (hostnames are not supported); in addition, exactly one DNS server IP address must be provided.

NOTE2: This API depends on VPN Always ON feature and it is not properly working on all Android devices. Make sure that this feature works on your device before using this API.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    boolean result = vpnPolicy.setAlwaysOn(profileName);
    if (result) {
        // Always ON mode was successfully configured
    } else {
        // Some error occured and Always ON was not set properly
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`createProfile(VpnAdminProfile)`

`getAlwaysOnProfile()`

public boolean **setDnsDomains** (String profileName, List<String> searchDomains)

Since: API level 5

API to set the VPN profile DNS search domains.

Parameters

profileName	The current profile name of the VPN profile.
searchDomains	The DNS search domains to be used for the VPN connection.

Returns

`true` if setting DNS search domains was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to change the DNS search domains in a VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    List listdnsDomains ;
    listdnsDomains.add("192.168.1.2");
    boolean success = vpnPolicy.setDnsDomains(profileName, listdnsDomains);
    if (success) {
        Log.d("VPNPolicy", "Setting the DNS search domain succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting the DNS search domain failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`getDnsDomains(String)`
`unlockCredentialStorage(String)`

public boolean setDnsServers (String profileName, List<String> dnsServers)

Since: API level 5

API to set the VPN profile DNS server IP addresses.

Parameters

profileName	The current profile name of the VPN profile.
dnsServers	The IP addresses of the DNS servers to be used for the VPN connection.

Returns

`true` if setting DNS servers is successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to set the DNS server IP addresses in a VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
A DNS server is any computer registered to join the Domain Name System. A DNS server runs special purpose networking software, features a public IP address, and contains a database of network names and addresses for other Internet hosts.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    List listDnsServer = null;
    listDnsServer.add("8.8.8.8");
    listDnsServer.add("7.7.7.7");
    boolean success = vpnPolicy.setDnsServers(profileName, listDnsServer);
    if (success) {
        Log.d("VPNPolicy", "Setting DNS server succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting DNS server failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`getDnsServers(String)`
`unlockCredentialStorage(String)`

public boolean setForwardRoutes (String profileName, List<String> routes)

Since: API level 5

API to set the VPN profile forward routes.

Parameters

profileName	The current profile name of the VPN profile.
routes	The IP of the forward routes in CIDR format to be used for the VPN connection.

Returns

`true` if setting forward routes was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to change the forward routes in a VPN profile provisioned on the device without any user interaction. To properly use this API, a VPN profile must already exist.
A forward route is the mechanism that forwards a network port from one network node to another. Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    List listRoutes = null;
    listRoutes.add("10.0.0.08");
    listRoutes.add("10.1.1.18");
    boolean success = vpnPolicy.setForwardRoutes(profileName, listRoutes);
    if (success) {
        Log.d("VPNPolicy", "Setting forward routes succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting forward routes failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`getForwardRoutes(String)`
`unlockCredentialStorage(String)`

public boolean setIPSecCaCertificate (String profileName, String certificate)

Since: API level 2

API to set the CA certificate for a VPN profile.

Parameters

profileName	The current profile name of the VPN profile.
certificate	The name of the CA certificate for the VPN profile.

Returns

`true` if setting the IP CA certificate was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can set the CA certificate of an L2TP/IPsec CRT VPN profile provisioned on the device without user interaction. To properly use this API, an L2TP/IPsec CRT profile must already exist. This certificate is picked up from the Android Keystore, hence the certificate must be installed on the device for the VPN connection to function properly. Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileCaCertificate = "certificateName";
    success = vpnPolicy.setIPSecCaCertificate(profileName, profileCaCertificate);
    if (success) {
        Log.d("VPNPolicy", "Setting VPN profile CA certificate has succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting VPN profile CA certificate has failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Dependency

the profile created must be of type L2TP/CRT

Multiuser Environment

Global Scope

See Also

`getIPSecCaCertificate(String)`
`unlockCredentialStorage(String)`

public boolean setIPSecPreSharedKey (String profileName, String psk)

Since: API level 2

API to set the pre-shared key for a VPN profile.

Parameters

profileName	The current profile name of the VPN profile.
psk	The pre-shared key of the VPN profile.

Returns

`true` if setting IP pre-shared key was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to change the pre-shared key of an L2TP/IPsec PSK VPN profile provisioned on the device without user interaction. To properly use this API, an L2TP/IPsec PSK profile must already exist. PSK is a shared secret that was previously shared between the two parts using some secure channel before it needs to be used.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profilePreSharedKey = "VpnPSK";
    boolean success = vpnPolicy.setIPSecPreSharedKey(profileName, profilePreSharedKey);
    if (success) {
        Log.d("VPNPolicy", "Setting VPN profile PSK has succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting VPN profile PSK has failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Dependency

the PSK profile created must be of type PSK

Multiuser Environment

Global Scope

See Also

`getIPSecPreSharedKey(String)`
`unlockCredentialStorage(String)`

public boolean setIPSecUserCertificate (String profileName, String certificate)

Since: API level 2

API to set the user certificate for a VPN profile.

Parameters

profileName	The current profile name of the VPN profile.
certificate	The name of the user certificate for the VPN profile.

Returns

`true` if setting the IP user certificate was successful, else `false`.

Throws

SecurityException	If caller does not have required permissionss

Usage

An administrator can use this API to set the user certificate of an L2TP/IPsec CRT VPN profile provisioned on the device without user interaction. To properly use this API, an L2TP/IPsec CRT profile must already exist. This certificate is picked up from the Android Keystore, hence the certificate must be installed on the device for the VPN connection to function properly. Notice that, until Knox API 29, the KeyStore must be unlocked. See unlockCredentialStorage(String).

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    // Creating L2TP/CRT profile
    // Keystore must be unlocked to create this kind of profile
    VpnAdminProfile profile = new VpnAdminProfile();
    profile.profileName = "L2TP CRT Profile";
    profile.serverName = "127.0.0.1";
    profile.vpnType = VpnAdminProfile.VPN_TYPE_L2TP_IPSEC_CRT;
    profile.userName = "username";
    profile.userPassword = "password";
    profile.L2TPSecret = "L2tpSecret";
    profile.IPSecCaCertificate = "CaCertificate";
    profile.IPSecUserCertificate = "UserCertificate";

    boolean success = vpnPolicy.createProfile(profile);
    if (success) {
        Log.d("VPNPolicy", "L2TP/IPsec CRT profile created!");
    } else {
        Log.d("VPNPolicy", "FAILED - L2TP/IPsec CRT profile not created.");
    }

    String profileName = "L2TP CRT Profile";
    String profileUserCertificate = "certificateName";
    success = vpnPolicy.setIPSecUserCertificate(profileName, profileUserCertificate);
    if (success) {
        Log.d("VPNPolicy", "Setting VPN profile user certificate has succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting VPN profile User certificate has failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Dependency

the profile created must be of type L2TP/RSA

Multiuser Environment

Global Scope

See Also

getIPSecUserCertificate(String)

unlockCredentialStorage(String)

public void setId (String profileName, String id)

Since: API level 2

API to set the Id for a VPN profile.

Parameters

profileName	The current profile name of the VPN profile.
id	The Id of the VPN profile to be configured.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to change the Id of a VPN profile provisioned on the device without user interaction. To properly use this API, a profile must already exist. The Id is the VPN internal reference used by Android to manage the profile. Notice that, until Knox API 29, the KeyStore must be unlocked. See unlockCredentialStorage(String).

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileId = "123456789";
    boolean success = vpnPolicy.setId(profileName, profileId);
    if (success) {
        Log.d("VPNPolicy", "Setting the VPN profile Id has succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting the VPN profile Id has failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Multiuser Environment

Global Scope

See Also

getId(String)

unlockCredentialStorage(String)

public boolean setIpSecIdentifier (String profileName, String ipSecIdentifier)

Since: API level 5

API to set the IP security (IPsec) identifier.

Parameters

profileName	The current profile name of the VPN profile.
ipSecIdentifier	The IP security identifier.

Returns

`true` if setting IPsec identifier was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to set the VPN IP security (IPsec) identifier. IPsec is a standard for providing security to IP protocols via encryption and/or authentication, typically employing both. Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 5
MDM 3.0

Multiuser Environment

Global Scope

See Also

`getIpSecIdentifier(String)`

public boolean setL2TPSecret (String profileName, boolean enabled, String secret)

Since: API level 2

API to set the secret for an L2TP VPN profile.

Parameters

profileName	The current profile name of the VPN profile.
enabled	`true` to enable use of secret on VPN Profile.
secret	The secret to be set on the VPN profile.

Returns

`true` if seting L2TP secret was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to set the secret of a L2TP VPN profile provisioned on the device without user interaction. To properly use this API, a VPN profile must already exist.
Notice that the the KeyStore must be unlocked. See unlockCredentialStorage(String).

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    boolean L2tpSecretEnabled = true;
    String L2tpSecret = "Secret";
    boolean success = vpnPolicy.setL2TPSecret(profileName, L2tpSecretEnabled,
L2tpSecret);
    if (success) {
        Log.d("VPNPolicy", "Setting VPN profile L2TP secret succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting VPN profile L2TP secret failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Dependency

the profile created must be L2TP type

Multiuser Environment

Global Scope

See Also

getL2TPSecret(String)
unlockCredentialStorage(String)

public boolean setOcspServerUrl (String profileName, String ocspServerUrl)

Since: API level 11

API to set the OCSP server URL.

Parameters

profileName	The current profile name of the VPN profile.
ocspServerUrl	The IP security identifier.

Returns

`true` on ocspServerUrl success, `false` on ocspServerUrl failure.

Throws

SecurityException	If caller does not have required permissions

Usage

Administrator can set Online Certificate Status Protocol (OCSP) server URL. If OCSP server URL is set, security gateway certificate is validated using OCSP during IKEv2 authentication. If the validation fails for some reason or if the certificate is reported to be revoked, connection will fail. If OCSP server URL is not set, or if it is set to empty string, OCSP validation is skipped.
OCSP is used only with IKEv2 RSA connection type. With other connection types this value is ignored.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "profileName";
    String ocspServerUrl = "http://ocsp.serverurl.org:8880";
    boolean ocspServer = vpnPolicy.setOcspServerUrl(profileName, ocspServerUrl);
    if (ocspServer) {
        Log.d("VPNPolicy", "Set Ocsp Server Url Success");
    } else {
        Log.d("VPNPolicy", "Set Ocsp Server Url Failed");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 11

MDM 5.0

Multiuser Environment

Global Scope

See Also

`getOcspServerUrl(String)`

public boolean setPPTPEncryptionEnabled (String profileName, boolean enabled)

Since: API level 2

API to enable or disable encryption for a VPN PPTP profile.

Parameters

profileName	The current profile name of the VPN profile.
enabled	`true` to enable encryption, `false` to disable encryption.

Returns

`true` if setting PPTP encryption was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to enable and disable encryption of a PPTP VPN profile provisioned on the device without user interaction. To properly use this API, a PPTP VPN profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Dependency

the profile created must be PPTP type

Multiuser Environment

Global Scope

See Also

`isPPTPEncryptionEnabled(String)`
`unlockCredentialStorage(String)`

public void **setProfileName** (String oldProfileName, String newProfileName)

Since: API level 2

API to change the profile name of a VPN profile.

Parameters

oldProfileName	The current profile name of the VPN profile.
newProfileName	The new profile name of the VPN profile.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can change the name of a VPN profile provisioned on the device without user interaction. To properly use this API, a profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

`unlockCredentialStorage(String)`

public void **setServerName** (String profileName, String serverName)

Since: API level 2

API to set the VPN server name.

Parameters

profileName	The current profile name of the VPN profile.
serverName	The hostname of the VPN server to be configured.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to change the server name of a VPN profile provisioned on the device without user interaction. To properly use this API, a profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See `unlockCredentialStorage(String)`.

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2

MDM 2.0

Multiuser Environment

Global Scope

See Also

`getServerName(String)`
`unlockCredentialStorage(String)`

public boolean setUserName (String profileName, String userName)

Since: API level 2

API to set the VPN user name.

Parameters

profileName	The current profile name of the VPN profile.
userName	The user name of the VPN profile to be configured

Returns

`true` if set user name success, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

Administrator can change the user name of a VPN provisioned on the device without any user interaction. In order to properly use this API a profile should have been created previously.
Notice that, until Knox API 29, the KeyStore must be unlocked. unlockCredentialStorage(String)

```
EnterpriseDeviceManager edm = (EnterpriseDeviceManager)
     getSystemService(EnterpriseDeviceManager.VPN_POLICY_SERVICE);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileUserName = "userName";
    boolean success = vpnPolicy.setUserName(profileName,profileUserName);
    if (success) {
        Log.d("VPNPolicy", "Set VPN profile user name is Success");
    } else {
        Log.d("VPNPolicy", "Set VPN profile user name  is Failure");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy","Exception: "+e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

getUserName(String)
unlockCredentialStorage(String)

public boolean setUserPassword (String profileName, String userPassword)

Since: API level 2

API to set the VPN user password.

Parameters

profileName	The current profile name of the VPN profile.
userPassword	The user password for the VPN profile.

Returns

`true` if setting user password was successful, else `false`.

Throws

SecurityException	If caller does not have required permissions

Usage

An administrator can use this API to change the user password of a VPN profile provisioned on the device without user interaction. To properly use this API, a profile must already exist.
Notice that, until Knox API 29, the KeyStore must be unlocked. See unlockCredentialStorage(String).

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
VpnPolicy vpnPolicy = edm.getVpnPolicy();
try {
    String profileName = "ProfileName";
    String profileUserPassword = "userPassword";
    boolean success = vpnPolicy.setUserPassword(profileName, profileUserPassword);
    if (success) {
        Log.d("VPNPolicy", "Setting VPN profile user password has succeeded.");
    } else {
        Log.d("VPNPolicy", "Setting VPN profile user password has failed.");
    }
} catch (SecurityException e) {
    Log.w("VPNPolicy", "Exception: " + e);
}
```

Permission

The use of this API requires the caller to have the "com.samsung.android.knox.permission.KNOX_VPN" permission which has the protection level of signature.

Since

API level 2
MDM 2.0

Multiuser Environment

Global Scope

See Also

getUserPassword(String)
unlockCredentialStorage(String)