iOS Configuration Profile Reference

Developer

Contents

Configuration Profile Key Reference

Note This document was previously titled *iPhone Configuration Profile Reference*.

A configuration profile is an XML file that allows you to distribute configuration information to iOS–based devices. If you need to configure a large number of devices or to provide lots of custom email settings, network settings, or certificates to a large number of devices, configuration profiles are an easy way to do it.

An iOS configuration profile contains a number of settings that you can specify, including:

Passcode policies

Restrictions on device features (disabling the camera, for example)

Wi-Fi settings

VPN settings

Email server settings

Exchange settings

LDAP directory service settings

CalDAV calendar service settings

Web clips

Credentials and keys

Advanced cellular network settings

Configuration profiles are in property list format, with data values stored in Base64 encoding. The `.plist` format can be read and written by any XML library.

There are four ways to deploy configuration profiles:

- By physically connecting the device as described in *iPhone Configuration Utility*
- In an email message
- On a webpage
- Using over-the air configuration as described in *Over-the-Air Profile Delivery and Configuration*

iOS also supports using encryption to protect the contents of profiles and guarantee data integrity. To learn about encrypted profile delivery, read *iPhone Configuration Utility* or *Over-the-Air Profile Delivery and Configuration*.

This document describes the keys in an iOS configuration profile and provides examples of the resulting XML payloads.

> **Note** Before you get started working with configuration profiles, you should create a skeleton configuration profile using iPhone Configuration Utility (iPCU). This provides a useful starting point that you can then modify as desired.

Configuration Profile Keys

At the top level, a profile property list contains the following keys:

Key	Type	Content
`HasRemovalPasscode`	Bool	Optional. Set to `true` if there is a removal passcode.
`IsEncrypted`	Bool	Optional. Set to `true` if the profile is encrypted.
`IsManaged`	Bool	Optional. Set to `true` if this profile was installed by the current MDM service.
`PayloadContent`	Array	Optional. Array of payload dictionaries. Not present if IsEncrypted is `true`.
`PayloadDescription`	String	Optional. A description of the profile, shown on the Detail screen for the profile. This should be descriptive enough to help the user decide whether to install the profile.
`PayloadDisplayName`	String	Optional. A human-readable name for the profile. This value is displayed on the Detail screen. It does not have to be unique.
`PayloadIdentifier`	String	A reverse-DNS style identifier (com.example.myprofile, for example) that identifies the profile. This string is used to determine whether a new profile should replace an existing one or should be added.
`PayloadOrganization`	String	Optional. A human-readable string containing the name of the organization that provided the profile.

Key	Type	Content
PayloadUUID	String	A globally unique identifier for the profile. The actual content is unimportant, but it must be globally unique. In Mac OS X, you can use uuidgen(1) to generate reasonable UUIDs.
PayloadRemoval-Disallowed	Bool	Optional. If present and set to true, the user cannot delete the profile (unless the profile has a removal password and the user provides it). If locked in this way, the profile can be replaced by a new version only if the profile identifier matches and the profile is signed by the same authority.
PayloadType	String	Currently, the only supported value is Configuration.
PayloadVersion	Number	The version number of the profile format. This describes the version of the configuration profile as a whole, not of the individual profiles within it. Currently, this value should always be 1.
SignerCertificates	Array	Optional. An array containing the certificate used to sign the profile, followed by any intermediate certificates, in DER-encoded X.509 format.

Keys in the payload dictionary are described in detail in the next section.

Payload Dictionary Keys Common to All Payloads

If a PayloadContent value is provided in a payload, each entry in the array is a dictionary representing a configuration payload. The following keys are common to all payloads:

Key	Type	Content
PayloadType	String	The payload type. The payload types are described in "Payload-Specific Property Keys" (page 6).
PayloadVersion	Number	The version number of the individual payload. A profile can consist of payloads with different version numbers. For example, changes to the VPN software in iOS might introduce a new payload version to support additional features, but Mail payload versions would not necessarily change in the same release.

Key	Type	Content
`PayloadIdentifier`	String	A reverse-DNS-style identifier for the specific payload. It is usually the same identifier as the root-level `PayloadIdentifier` value with an additional component appended.
`PayloadUUID`	String	A globally unique identifier for the payload. The actual content is unimportant, but it must be globally unique. In Mac OS X, you can use `uuidgen(1)` to generate reasonable UUIDs.
`PayloadDisplayName`	String	Optional. A human-readable name for the profile payload. This name is displayed on the Detail screen. It does not have to be unique.
`PayloadDescription`	String	Optional. A human-readable description of this payload. This description is shown on the Detail screen.
`PayloadOrganization`	String	Optional. A human-readable string containing the name of the organization that provided the profile. The payload organization for a payload need not match the payload organization in the enclosing profile.

Payload-Specific Property Keys

In addition to the standard payload keys (described in "Payload Dictionary Keys Common to All Payloads" (page 5)), each payload type contains that are specific to that payload type. The sections that follow describe those payload-specific keys.

Profile Removal Password Payload

The Removal Password payload is designated by specifying `com.apple.profileRemovalPassword` value as the `PayloadType` value.

A password removal policy payload provides a password to allow users to remove a locked configuration profile from the device. If this payload is present and has a password value set, the device asks for the password when the user taps a profile's Remove button. This payload is encrypted with the rest of the profile.

Key	Type	Value
`RemovalPassword`	String	Optional. Specifies the removal password for the profile.

Passcode Policy Payload

The Passcode Policy payload is designated by specifying `com.apple.mobiledevice.passwordpolicy` as the `PayloadType` value.

The presence of this payload type prompts device to present the user with an alphanumeric passcode entry mechanism, which allows the entry of arbitrarily long and complex passcodes.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
`allowSimple`	Bool	Optional. Default `true`. Determines whether a simple passcode is allowed. A simple passcode is defined as containing repeated characters, or increasing/decreasing characters (such as 123 or CBA). Setting this value to `false` is synonymous to setting minComplexChars to "1".
`forcePIN`	Bool	Optional. Default NO. Determines whether the user is forced to set a PIN. Simply setting this value (and not others) forces the user to enter a passcode, without imposing a length or quality.
`maxFailedAttempts`	Number	Optional. Default 11. Allowed range [2...11]. Specifies the number of allowed failed attempts to enter the passcode at the device's lock screen. Once this number is exceeded, the device is locked and must be connected to its designated iTunes in order to be unlocked.
`maxInactivity`	Number	Optional. Default Infinity. Specifies the number of minutes for which the device can be idle (without being unlocked by the user) before it gets locked by the system. Once this limit is reached, the device is locked and the passcode must be entered.
`maxPINAgeInDays`	Number	Optional. Default Infinity. Specifies the number of days for which the passcode can remain unchanged. After this number of days, the user is forced to change the passcode before the device is unlocked.
`minComplexChars`	Number	Optional. Default 0. Specifies the minimum number of complex characters that a passcode must contain. A "complex" character is a character other than a number or a letter, such as &%$#.

Key	Type	Value
minLength	Number	Optional. Default 0. Specifies the minimum overall length of the passcode. This parameter is independent of the also optional minComplexChars argument.
requireAlphanumeric	Bool	Optional. Default NO. Specifies whether the user must enter alphabetic characters ("abcd"), or if numbers are sufficient.
pinHistory	Number	Optional. When the user changes the passcode, it has to be unique within the last N entries in the history. Minimum value is 1, maximum value is 50.
manualFetching-WhenRoaming	Bool	Optional. If set, all push operations will be disabled when roaming. The user has to manually fetch new data.
maxGracePeriod	Number	Optional. The maximum grace period, in minutes, to unlock the phone without entering a passcode. Default is 0, that is no grace period, which requires a passcode immediately.

Email Payload

The email payload is designated by specifying `com.apple.mail.managed` as the `PayloadType` value.

An email payload creates an email account on the device.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
EmailAccount-Description	String	Optional. A user-visible description of the email account, shown in the Mail and Settings applications.
EmailAccountName	String	Optional. The full user name for the account. This is the user name in sent messages, etc.
EmailAccountType	String	Allowed values are `EmailTypePOP` and `EmailTypeIMAP`. Defines the protocol to be used for that account.
EmailAddress	String	Designates the full email address for the account. If not present in the payload, the device prompts for this string during profile installation.
IncomingMailServer-Authentication	String	Designates the authentication scheme for incoming mail. Allowed values are `EmailAuthPassword` and `EmailAuthNone`.

Key	Type	Value
`IncomingMailServer–HostName`	String	Designates the incoming mail server host name (or IP address).
`IncomingMailServer–PortNumber`	Number	Optional. Designates the incoming mail server port number. If no port number is specified, the default port for a given protocol is used.
`IncomingMailServer–UseSSL`	Bool	Optional. Default `true`. Designates whether the incoming mail server uses SSL for authentication.
`IncomingMailServer–Username`	String	Designates the user name for the email account, usually the same as the email address up to the @ character. If not present in the payload, and the account is set up to require authentication for incoming email, the device will prompt for this string during profile installation.
`IncomingPassword`	String	Optional. Password for the Incoming Mail Server. Use only with encrypted profiles.
`OutgoingPassword`	String	Optional. Password for the Outgoing Mail Server. Use only with encrypted profiles.
`OutgoingPasswordSame–AsIncomingPassword`	Bool	Optional. If set, the user will be prompted for the password only once and it will be used for both outgoing and incoming mail.
`OutgoingMailServer–Authentication`	String	Designates the authentication scheme for outgoing mail. Allowed values are EmailAuthPassword and EmailAuthNone.
`OutgoingMailServer–HostName`	String	Designates the outgoing mail server host name (or IP address).
`OutgoingMailServer–PortNumber`	Number	Optional. Designates the outgoing mail server port number. If no port number is specified, ports 25, 587 and 465 are used, in this order.
`OutgoingMailServer–UseSSL`	Bool	Optional. Default Yes. Designates whether the outgoing mail server uses SSL for authentication.
`OutgoingMailServer–Username`	String	Designates the user name for the email account, usually the same as the email address up to the @ character. If not present in the payload, and the account is set up to require authentication for outgoing email, the device prompts for this string during profile installation.

Key	Type	Value
PreventMove	Bool	Optional. Default false. If true, messages may not be moved out of this email account into another account. Also prevents forwarding or replying from a different account than the message was originated from. **Availability:** Available in iOS 5.0 and later.
PreventAppSheet	Bool	Optional. Default false. If true, this account is not available for sending mail in third-party applications. **Availability:** Available in iOS 5.0 and later.
SMIMEEnabled	Bool	Optional. Default false. If true, this account supports S/MIME. **Availability:** Available in iOS 5.0 and later.
SMIMESigning-CertificateUUID	String	Optional. The PayloadUUID of the identity certificate used to sign messages sent from this account. **Availability:** Available in iOS 5.0 and later.
SMIMEEncryption-CertificateUUID	String	Optional. The PayloadUUID of the identity certificate used to decrypt messages coming into this account. **Availability:** Available in iOS 5.0 and later.

Web Clip Payload

The Web Clip payload is designated by specifying com.apple.webClip.managed as the PayloadType value.

A Web Clip payload provides a web clipping on the user's home screen as though the user had saved a bookmark to the home screen.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
URL	String	The URL that the Web Clip should open when clicked. The URL must begin with HTTP or HTTPS or it won't work.
Label	String	The name of the Web Clip as displayed on the Home screen.

Key	Type	Value
Icon	Data	Optional. A PNG icon to be shown on the Home screen. Should be 59 x 60 pixels in size. If not specified, a white square will be shown.
IsRemovable	Bool	Optional. If No, the user cannot remove the Web Clip, but it will be removed if the profile is deleted.

Restrictions Payload

The Restrictions payload is designated by specifying `com.apple.applicationaccess` as the `PayloadType` value.

A Restrictions payload allows the administrator to restrict the user from doing certain things with the device, such as using the camera.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
allowAppInstallation	Bool	Optional. When false, the App Store is disabled and its icon is removed from the Home screen. Users are unable to install or update their applications.
AllowAssistant	Bool	Optional. When `false`, disables Siri. Defaults to `true`.
allowCamera	Bool	Optional. When false, the camera is completely disabled and its icon is removed from the Home screen. Users are unable to take photographs.
allowExplicitContent	Bool	Optional. When false, explicit music or video content purchased from the iTunes Store is hidden. Explicit content is marked as such by content providers, such as record labels, when sold through the iTunes Store.
allowScreenShot	Bool	Optional. When false, users are unable to save a screenshot of the display.
allowYouTube	Bool	Optional. When false, the YouTube application is disabled and its icon is removed from the Home screen.
allowiTunes	Bool	Optional. When false, the iTunes Music Store is disabled and its icon is removed from the Home screen. Users cannot preview, purchase, or download content.

Key	Type	Value
forceITunesStore-PasswordEntry	Bool	Optional. When true, forces user to enter their iTunes password for each transaction. **Availability:** Available in iOS 5.0 and later.
allowSafari	Bool	Optional. When false, the Safari web browser application is disabled and its icon removed from the Home screen. This also prevents users from opening web clips.
allowUntrusted-TLSPrompt	Bool	Optional. When false, automatically rejects untrusted HTTPS certificates without prompting the user. **Availability:** Available in iOS 5.0 and later.
allowCloudBackup	Bool	Optional. When false, disables backing up the device to iCloud. **Availability:** Available in iOS 5.0 and later.
allowCloudDocument-Sync	Bool	Optional. When false, disables document and key-value syncing to iCloud. **Availability:** Available in iOS 5.0 and later.
allowPhotoStream	Bool	Optional. When false, disables Photo Stream. **Availability:** Available in iOS 5.0 and later.

LDAP Payload

The LDAP payload is designated by specifying com.apple.ldap.account as the PayloadType value.

An LDAP payload provides information about an LDAP server to use, including account information if required, and a set of LDAP search policies to use when querying that LDAP server.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
LDAPAccount-Description	String	Optional. Description of the account.
LDAPAccountHostName	String	The host.
LDAPAccountUseSSL	Bool	Whether or not to use SSL.
LDAPAccountUserName	String	Optional. The username.

Key	Type	Value
LDAPAccountPassword	String	Optional. Use only with encrypted profiles.
LDAPSearchSettings	Dictionary	Top level container object. Can have many of these for one account. Should have at least one for the account to be useful. Each LDAPSearchSettings object represents a node in the LDAP tree to start searching from, and tells what scope to search in (the node, the node plus one level of children, or the node plus all levels of children).
LDAPSearchSetting–Description	String	Optional. Description of this search setting.
LDAPSearchSetting–SearchBase	String	Conceptually, the path to the node to start a search at. For example: ou=people,o=example corp
LDAPSearchSetting–Scope	String	Defines what recursion to use in the search. Can be one of the following 3 values: LDAPSearchSettingScopeBase: Just the immediate node pointed to by SearchBase LDAPSearchSettingScopeOneLevel: The node plus its immediate children. LDAPSearchSettingScopeSubtree: The node plus all children, regardless of depth.

CalDAV Payload

The CalDAV payload is designated by specifying com.apple.caldav.account as the PayloadType value.

A CalDAV payload adds a CalDAV account to the user's calendars list.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
CalDAVAccountDescription	String	Optional. Description of the account.
CalDAVHostName	String	The server address
CalDAVUsername	String	The user's login name.

Key	Type	Value
CalDAVPassword	String	Optional. The user's password
CalDAVUseSSL	Bool	Whether or not to use SSL.
CalDAVPort	Number	Optional. The port on which to connect to the server.
CalDAVPrincipalURL	String	Optional. The base URL to the user's calendar.

Calendar Subscription Payload

The CalSub payload is designated by specifying `com.apple.subscribedcalendar.account` as the `PayloadType` value.

A CalSub payload adds a subscribed calendar to the user's calendars list.

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
SubCalAccountDescription	String	Optional. Description of the account.
SubCalAccountHostName	String	The server address.
SubCalAccountUsername	String	The user's login name
SubCalAccountPassword	String	The user's password.
SubCalAccountUseSSL	Bool	Whether or not to use SSL.

SCEP Payload

The SCEP (Simple Certificate Enrollment Protocol) payload is designated by specifying `com.apple.encrypted-profile-service` as the `PayloadType` value.

An SCEP payload associates the phone with an SCEP server, as described in *Over-the-Air Profile Delivery and Configuration* .

In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
URL	String	The SCEP URL. See *Over-the-Air Profile Delivery and Configuration* for more information about SCEP.

Key	Type	Value
Name	String	Optional. Any string that is understood by the SCEP server. For example, it could be a domain name like example.org. If a certificate authority has multiple CA certificates this field can be used to distinguish which is required.
Subject	Array	Optional. The representation of a X.500 name represented as an array of OID and value. For example, /C=US/O=Apple Inc./CN=foo/1.2.5.3=bar, which would translate to: [[["C", "US"]], [["O", "Apple Inc."]], ..., [["1.2.5.3", "bar"]]] OIDs can be represented as dotted numbers, with shortcuts for country (C), locality (L), state (ST), organization (O), organizational unit (OU), and common name (CN).
Challenge	String	Optional. A pre-shared secret.
Keysize	Number	Optional. The key size in bits, either 1024 or 2048.
Key Type	String	Optional. Currently always "RSA".
Key Usage	Number	Optional. A bitmask indicating the use of the key. 1 is signing, 4 is encryption, 5 is both signing and encryption. Some CAs, such as Windows CA, support only encryption or signing, but not both at the same time.

SubjectAltName Dictionary Keys

The SCEP payload can specify an optional SubjectAltName dictionary that provides values required by the CA for issuing a certificate. You can specify a single string or an array of strings for each key.

The values you specify depend on the CA you're using, but might include DNS name, URL, or email values. For an example, see "Sample Configuration Profile" (page 23) or read *Over-the-Air Profile Delivery and Configuration*.

GetCACaps Dictionary Keys

If you add a dictionary with the key GetCACaps, the device uses the strings you provide as the authoritative source of information about the capabilities of your CA. Otherwise, the device queries the CA for GetCACaps and uses the answer it gets in response.If the CA doesn't respond, the device defaults to GET 3DES and SHA-1 requests. For more information, read *Over-the-Air Profile Delivery and Configuration*.

APN Payload

The APN (Access Point Name) payload is designated by specifying `com.apple.apn.managed` as the `PayloadType` value. In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
`DefaultsData`	Dictionary	This dictionary contains two key/value pairs.
`DefaultsDomainName`	String	The only allowed value is `com.apple.managedCarrier`.
`apns`	Array	This array contains an arbitrary number of dictionaries, each describing an APN configuration, with the key/value pairs below.
`apn`	String	This string specifies the Access Point Name.
`username`	String	This string specifies the user name for this APN. If it is missing, the device prompts for it during profile installation.
`password`	Data	Optional. This data represents the password for the user for this APN. For obfuscation purposes, the password is encoded. If it is missing from the payload, the device prompts for the password during profile installation.
`proxy`	String	Optional. The IP address or URL of the APN proxy.
`proxyPort`	Number	Optional. The port number of the APN proxy.

Exchange Payload

The Exchange payload is designated by specifying `com.apple.eas.account` as the `PayloadType` value. This payload creates a Microsoft Exchange account on the device. In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
`EmailAddress`	String	Specifies the full email address for the account. If not present in the payload, the device prompts for this string during profile installation.
`Host`	String	Specifies the Exchange server host name (or IP address).
`SSL`	Bool	Optional. Default YES. Specifies whether the Exchange server uses SSL for authentication.

Key	Type	Value
`UserName`	String	This string specifies the user name for this Exchange account. If missing, the devices prompts for it during profile installation.
`Password`	String	Optional. The password of the account. Use only with encrypted profiles.
`Certificate`	NSData blob	Optional. For accounts that allow authentication via certificate, a .p12 identity certificate in NSData blob format.
`CertificateName`	String	Optional. Specifies the name or description of the certificate.
`CertificatePassword`	data	Optional. The password necessary for the p12 identity certificate. Use only with encrypted profiles.
`PreventMove`	Bool	Optional. Default `false`. If set to `true`, messages may not be moved out of this email account into another account. Also prevents forwarding or replying from a different account than the message was originated from. **Availability:** Available in iOS 5.0 and later.
`PreventAppSheet`	Bool	Optional. Default `false`. If set to `true`, this account will not be available for sending mail in third party applications. **Availability:** Available in iOS 5.0 and later.
`PayloadCertificate-UUID`	String	UUID of the certificate payload to use for the identity credential. If this field is present, the `Certificate` field is not used. **Availability:** Available in iOS 5.0 and later.
`SMIMEEnabled`	Bool	Optional. Default `false`. If set to `true`, this account supports S/MIME. **Availability:** Available in iOS 5.0 and later.
`SMIMESigning-CertificateUUID`	String	Optional. The `PayloadUUID` of the identity certificate used to sign messages sent from this account. **Availability:** Available in iOS 5.0 and later.
`SMIMEEncryption-CertificateUUID`	String	Optional. The `PayloadUUID` of the identity certificate used to decrypt messages coming into this account. **Availability:** Available in iOS 5.0 and later.

Note Note: As with VPN and Wi-Fi configurations, it is possible to associate an SCEP credential with an Exchange configuration via the `PayloadCertificateUUID` key.

VPN Payload

The VPN payload is designated by specifying `com.apple.vpn.managed` as the `PayloadType` value. In addition to the settings common to all payload types, the VPN payload defines the following keys.

Key	Type	Value
`UserDefinedName`	String	Description of the VPN connection displayed on the device.
`OverridePrimary`	Bool	Specifies whether to send all traffic through the VPN interface. If true, all network traffic is sent over VPN.
`VPNType`	String	Determines the settings available in the payload for this type of VPN connection. It can have three possible values: "L2TP", "PPTP", or "IPSec", representing L2TP, PPTP and Cisco IPSec respectively.

There are two possible dictionaries present at the top level, under the keys "PPP" and "IPSec". The keys inside these two dictionaries are described below, along with the VPNType value under which the keys are used.

PPP Dictionary Keys

The following elements are for VPN payloads of type PPP.

Key	Type	Value
`AuthName`	String	The VPN account user name. Used for L2TP and PPTP.
`AuthPassword`	String	Optional. Only visible if TokenCard is false. Used for L2TP and PPTP.
`TokenCard`	Boolean	Whether to use a token card such as an RSA SecurID card for connecting. Used for L2TP.
`CommRemoteAddress`	String	IP address or host name of VPN server. Used for L2TP and PPTP.
`AuthEAPPlugins`	Array	Only present if RSA SecurID is being used, in which case it has one entry, a string with value "EAP-RSA". Used for L2TP and PPTP.
`AuthProtocol`	Array	Only present if RSA SecurID is being used, in which case it has one entry, a string with value "EAP". Used for L2TP and PPTP.

Key	Type	Value
CCPMPPE40Enabled	Boolean	See discussion under CCPEnabled. Used for PPTP.
CCPMPPE128Enabled	Boolean	See discussion under CCPEnabled. Used for PPTP.
CCPEnabled	Boolean	Enables encryption on the connection. If this key and CCPMPPE40Enabled are true, represents automatic encryption level; if this key and CCPMPPE128Enabled are true, represents maximum encryption level. If no encryption is used, then none of the CCP keys are true. Used for PPTP.

IPSec Dictionary Keys

The following elements are for VPN payloads of type IPSec.

Key	Type	Value
RemoteAddress	String	IP address or host name of the VPN server. Used for Cisco IPSec.
AuthenticationMethod	String	Either "SharedSecret" or "Certificate". Used for L2TP and Cisco IPSec.
XAuthName	String	User name for VPN account. Used for Cisco IPSec.
XAuthEnabled	Integer	1 if XAUTH is ON, 0 if it is OFF. Used for Cisco IPSec.
LocalIdentifier	String	Present only if AuthenticationMethod = SharedSecret. The name of the group to use. If Hybrid Authentication is used, the string must end with "[hybrid]". Used for Cisco IPSec.
LocalIdentifierType	String	Present only if AuthenticationMethod = SharedSecret. The value is "KeyID". Used for L2TP and Cisco IPSec.
SharedSecret	Data	The shared secret for this VPN account. Only present if AuthenticationMethod = SharedSecret. Used for L2TP and Cisco IPSec.
PayloadCertificate-UUID	String	The UUID of the certificate to use for the account credentials. Only present if AuthenticationMethod = Certificate. Used for Cisco IPSec.
PromptForVPNPIN	Bool	Tells whether to prompt for a PIN when connecting. Used for Cisco IPSec.

Wi-Fi Payload

The Wi-Fi payload is designated by specifying `com.apple.wifi.managed` as the `PayloadType` value. This describes version `0` (`PayloadVersion` value).

In addition to the settings common to all payload types, the payload defines the following keys.

Key	Type	Value
`SSID_STR`	String	SSID of the Wi-Fi network to be used.
`HIDDEN_NETWORK`	Bool	Besides SSID, the device uses information such as broadcast type and encryption type to differentiate a network. By default (`false`), it is assumed that all configured networks are open or broadcast. To specify a hidden network, must be `true`.
`AutoJoin`	Bool	Optional. Default `true`. If `true`, the network is auto-joined. If `false`, the user has to tap the network name to join it. **Availability:** Available in iOS 5.0 and later.
`EncryptionType`	String	The possible values are `WEP`, `WPA`, `Any`, and `None`. `WPA` corresponds to WPA and WPA2 and applies to both encryption types. Make sure that these values exactly match the capabilities of the network access point. If you're unsure about the encryption type, or would prefer that it apply to all encryption types, use the value `Any`. **Availability:** Available in iOS 4.0 and later; the `None` value is available in iOS 5.0 and later.
`Password`	String	Optional. The absence of a password doesn't prevent the network from being added to the list of known networks. The user is eventually prompted to provide the password when connecting to that network.
`ProxyType`	String	Optional. Valid values are `None`, `Manual`, and `Auto`. **Availability:** Available in iOS 5.0 and later.

If the `ProxyType` field is set to `Manual`, the following fields must also be provided:

Key	Type	Value
`ProxyServer`	String	The proxy server's network address.
`ProxyServerPort`	Integer	The proxy server's port.
`ProxyUsername`	String	Optional. The username used to authenticate to the proxy server.

Key	Type	Value
ProxyPassword	String	Optional. The password used to authenticate to the proxy server.

If the ProxyType field is set to Auto, the following fields must also be provided:

Key	Type	Value
ProxyPACURL	String	The URL of the PAC file that defines the proxy configuration.

For 802.1X enterprise networks, the EAP Client Configuration Dictionary must be provided.

EAPClientConfiguration Dictionary

In addition to the standard encryption types, it is possible to specify an enterprise profile for a given network via the "EAPClientConfiguration" key. If present, its value is a dictionary with the following keys.

Key	Type	Value
UserName	String	Optional. Unless you know the exact user name, this property won't appear in an imported configuration. Users can enter this information when they authenticate.
AcceptEAPTypes	Array of integers.	The following EAP types are accepted: 13 = TLS 17 = LEAP 21 = TTLS 25 = PEAP 43 = EAP-FAST
PayloadCertificate-AnchorUUID	Array of strings	Optional. Identifies the certificates to be trusted for this authentication. Each entry must contain the UUID of a certificate payload. Use this key to prevent the device from asking the user if the listed certificates are trusted. Dynamic trust (the certificate dialogue) is disabled if this property is specified, unless TLSAllowTrustExceptions is also specified with the value true.

Key	Type	Value
TLSTrustedServer-Names	Array of strings	Optional. This is the list of server certificate common names that will be accepted. You can use wildcards to specify the name, such as wpa.*.example.com. If a server presents a certificate that isn't in this list, it won't be trusted. Used alone or in combination with TLSTrustedCertificates, the property allows someone to carefully craft which certificates to trust for the given network, and avoid dynamically trusted certificates. Dynamic trust (the certificate dialogue) is disabled if this property is specified, unless TLSAllowTrustExceptions is also specified with the value true.
TLSAllowTrust-Exceptions	Bool	Optional. Allows/disallows a dynamic trust decision by the user. The dynamic trust is the certificate dialogue that appears when a certificate isn't trusted. If this is false, the authentication fails if the certificate isn't already trusted. See PayloadCertificateAnchorUUID and TLSTrustedNames above. The default value of this property is true unless either PayloadCertificateAnchorUUID or TLSTrustedServerNames is supplied, in which case the default value is false.
TTLSInner-Authentication	String	Optional. This is the inner authentication used by the TTLS module. The default value is "MSCHAPv2". Possible values are "PAP", "CHAP", "MSCHAP", and "MSCHAPv2".
OuterIdentity	String	Optional. This key is only relevant to TTLS, PEAP, and EAP-FAST. This allows the user to hide his or her identity. The user's actual name appears only inside the encrypted tunnel. For example, it could be set to "anonymous" or "anon", or "anon@mycompany.net". It can increase security because an attacker can't see the authenticating user's name in the clear.

EAP-Fast Support

The EAP-FAST module uses the following properties in the EAPClientConfiguration dictionary.

Key	Type	Value
EAPFASTUsePAC	Boolean	Optional.
EAPFASTProvisionPAC	Boolean	Optional.
EAPFASTProvisionPACAnonymously	Boolean	Optional.

These keys are hierarchical in nature: if EAPFASTUsePAC is false, the other two properties aren't consulted. Similarly, if EAPFASTProvisionPAC is false, EAPFASTProvisionPACAnonymously isn't consulted.

If EAPFASTUsePAC is false, authentication proceeds much like PEAP or TTLS: the server proves its identity using a certificate each time.

If EAPFASTUsePAC is true, then an existing PAC is used if present. The only way to get a PAC on the device currently is to allow PAC provisioning. So, you need to enable EAPFASTProvisionPAC, and if desired, EAPFASTProvisionPACAnonymously. EAPFASTProvisionPACAnonymously has a security weakness: it doesn't authenticate the server so connections are vulnerable to a man-in-the-middle attack.

Certificates

As with VPN configurations, it is possible to associate a certificate identity configuration with a Wi-Fi configuration. This is useful when defining credentials for a secure enterprise network. To associate an identity, specify its payload UUID via the "PayloadCertificateUUID" key.

Key	Type	Value
PayloadCertificateUUID	String	UUID of the certificate payload to use for the identity credential.

Sample Configuration Profile

The following is a sample configuration profile containing an SCEP payload.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Inc//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
    <dict>
        <key>PayloadVersion</key>
        <integer>1</integer>
```

```
        <key>PayloadUUID</key>
        <string>Ignored</string>
        <key>PayloadType</key>
        <string>Configuration</string>
        <key>PayloadIdentifier</key>
        <string>Ignored</string>
        <key>PayloadContent</key>
        <array>
            <dict>
                <key>PayloadContent</key>
                <dict>
                    <key>URL</key>
                    <string>https://scep.example.com/scep</string>
                    <key>Name</key>
                    <string>EnrollmentCAInstance</string>
                    <key>Subject</key>
                    <array>
                        <array>
                            <array>
                                <string>O</string>
                                <string>Example, Inc.</string>
                            </array>
                        </array>
                        <array>
                            <array>
                                <string>CN</string>
                                <string>User Device Cert</string>
                            </array>
                        </array>
                    </array>
                    <key>Challenge</key>
                    <string>...</string>
                    <key>Keysize</key>
                    <integer>1024</integer>
```

```
                    <key>Key Type</key>
                    <string>RSA</string>
                    <key>Key Usage</key>
                    <integer>5</integer>
                </dict>
                <key>PayloadDescription</key>
                <string>Provides device encryption identity</string>
                <key>PayloadUUID</key>
                <string>fd8a6b9e-0fed-406f-9571-8ec98722b713</string>
                <key>PayloadType</key>
                <string>com.apple.security.scep</string>
                <key>PayloadDisplayName</key>
                <string>Encryption Identity</string>
                <key>PayloadVersion</key>
                <integer>1</integer>
                <key>PayloadOrganization</key>
                <string>Example, Inc.</string>
                <key>PayloadIdentifier</key>
                <string>com.example.profileservice.scep</string>
            </dict>
        </array>
    </dict>
</plist>
```

Document Revision History

This table describes the changes to *iOS Configuration Profile Reference*.

Date	Notes
2011-10-17	Removed extraneous iCloud key.
2011-10-12	Updated for iOS 5.0.
2011-03-08	Retitled document.
2010-09-21	Fixed typographical errors.
2010-08-03	New document that describes the property list keys used in iOS configuration profiles.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010