Configuration Profile Reference

Developer

Contents

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Tables

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Configuration Profile Key Reference

Note: This document was previously titled *iPhone Configuration Profile Reference*. It now supports both iOS and OS X.

A configuration profile is an XML file that allows you to distribute configuration information. 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.

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

Restrictions on device features

Wi-Fi settings

VPN settings

Email server settings

Exchange settings

LDAP directory service settings

CalDAV calendar service settings

Web clips

Credentials and keys

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 five ways to deploy configuration profiles:

- Using Apple Configurator (iOS only)
- In an email message
- On a webpage
- Using over-the air configuration as described in *Over-the-Air Profile Delivery and Configuration*
- Over the air using a Mobile Device Management Server

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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

Devices running iOS 5 and later can be designated as supervised when preparing it for deployment with Apple Configurator or using the Device Enrollment Program. When a device is supervised, you are granted additional control over its configuration. For more information about Apple Configurator, see Apple's iPad in Business page.

This document describes the keys in a 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. 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`	Boolean	Optional. Set to `true` if there is a removal passcode.
`IsEncrypted`	Boolean	Optional. Set to `true` if the profile is encrypted.
`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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Content
`PayloadUUID`	String	A globally unique identifier for the profile. The actual content is unimportant, but it must be globally unique. In OS X, you can use `uuidgen` to generate reasonable UUIDs.
`PayloadRemoval-Disallowed`	Boolean	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).
`PayloadType`	String	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 be `1`.
`PayloadScope`	String	Determines if the profile should be installed for the system or the user. In many cases, it determines the location of the certificate items, such as keychains. Though it is not possible to declare different payload scopes, payloads, like VPN, may automatically install their items in both scopes if needed. Legal values are `System` and `User`. **Availability:** Available in OS X v10.8 and later.
`RemovalDate`	date	Optional. The date on which the profile will be automatically removed.
`DurationUntilRemoval`	float	Optional. Number of seconds until the profile is automatically removed. If the `RemovalDate` keys is present, its value is used instead of this one.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Content
`ConsentText`	Dictionary	Optional. This dictionary's keys must be locale strings that contain a canonicalized IETF BCP 47 language identifier. Additionally, the key `default` may be present to provide the default localization. The system chooses a localized version in the order of preference specified by the user (OS X) or based on the user's current language setting (iOS). If no exact match is found, the default localization is used. If there is no default localization, the `"en"` localization is used. If there is no `"en"` localization, then the first available localization is used. You should provide a default localization. No warning will be displayed if the user's locale does not match any of the localizations in the consentText dictionary.

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 9).
`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.
`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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Content
`PayloadUUID`	String	A globally unique identifier for the payload. The actual content is unimportant, but it must be globally unique. In OS X, you can use `uuidgen` 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 8)), each payload type contains keys that are specific to that payload type. The sections that follow describe those payload-specific keys.

AirPlay Payload

The AirPlay payload is designated by specifying `com.apple.airplay` as the `PayloadType` value.

This payload is supported only on iOS 7.0 and later.

Key	Type	Value
`Whitelist`	Array of dictionaries	Optional. Supervised only (ignored otherwise). If present, only AirPlay destinations present in this list are available to the device. The dictionary format is described below.
`Passwords`	Array of dictionaries	Optional. If present, sets passwords for known AirPlay destinations. The dictionary format is described below.

Each entry in the `Whitelist` array is a dictionary that can contain the following fields:

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
DeviceID	String	The Device ID of the AirPlay destination, in the format xx:xx:xx:xx:xx:xx. This field is not case sensitive.

Each entry in the Passwords array is a dictionary that can contain the following fields:

Key	Type	Value
DeviceName	String	The name of the AirPlay destination.
Password	String	The password for the AirPlay destination.

AirPrint Payload

The AirPrint payload adds AirPrint printers to the user's AirPrint printer list. This makes it easier to support environments where the printers and the devices are on different subnets. An AirPrint payload is designated by specifying com.apple.airprint as the PayloadType value.

This payload is supported only on iOS 7.0 and later.

Key	Type	Value
AirPrint	Array of dictionaries	An array of AirPrint printers that should always be shown.

Each dictionary in the AirPrint array must contain the following keys and values:

Key	Type	Value
IPAddress	String	The IP Address of the AirPrint destination.
ResourcePath	String	The Resource Path associated with the printer. This corresponds to the rp parameter of the _ipps.tcp Bonjour record. For example: printers/Canon_MG5300_series printers/Xerox_Phaser_7600 ipp/print Epson_IPP_Printer

Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

APN Payload

The APN (Access Point Name) payload is designated by specifying `com.apple.apn.managed` as the `PayloadType` value.

In iOS 7 and later, the APN payload is deprecated in favor of the Cellular payload.

The APN Payload is not supported in OS X.

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.

Per-App VPN Payload

The Per-App VPN payload, is used for configuring add-on VPN software. This payload should not be confused with the standard VPN payload, described in "VPN Payload" (page 36).

This payload is supported only in iOS 7.0 and later and OS X v10.9 and later.

The VPN payload is designated by specifying `com.apple.vpn.managed.applayer` as the `PayloadType` value. The Per-App VPN payload supports all of the keys described in "VPN Payload" (page 36) plus the following additional keys:

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
VPNUUID	String	A globally-unique identifier for this VPN configuration. This identifier is used to configure apps so that they use the Per-App VPN service for all of their network communication. See "App-to-Per-App VPN Mapping" (page 12) and the managed app section of *Mobile Device Management Protocol Reference* to learn how to specify which apps should use this Per-App VPN service.
SafariDomains	Array	This array contains strings, each of which is a domain that should trigger this VPN connection in Safari. The rule matching behavior is as follows: • Before being matched against a host, all leading and trailing dots are stripped from the domain string. For example, if the domain string is ".com" the domain string used to match is "com". • Each label in the domain string must match an entire label in the host string. For example, a domain of "apple.com" matches "www.apple.com", but not "foo.badapple.com". • Domain strings with only one label must match the entire host string. For example, a domain of "com" matches "com", not "www.apple.com".
OnDemandMatch–AppEnabled	Boolean	If true, the Per-App VPN connection starts automatically when apps linked to this Per-App VPN service initiate network communication. If false, the Per-App VPN connection must be started manually by the user before apps linked to this Per-App VPN service can initiate network communication. If this key is not present, the value of the OnDemandEnabled key is used to determine the status of Per-App VPN On Demand.

App-to-Per-App VPN Mapping

The App-to-Per-App mapping payload is designated by specifying com.apple.vpn.managed.appmapping as the PayloadType value.

This payload is supported only in OS X v10.9 and later. It is not supported in iOS.

Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
AppLayerVPNMapping	Array of dictionaries	An array of mapping dictionaries.

Each dictionary in the array can contain the following keys:

Key	Type	Value
Identifier	String	The app's bundle ID.
VPNUUID	String	The VPNUUID of the Per-App VPN defined in a Per-App VPN payload.

App Lock Payload

The App Lock payload is designated by specifying `com.apple.app.lock` as the `PayloadType` value. Only one of this payload type can be installed at any time. This payload can be installed only on a Supervised device.

By installing an app lock payload, the device is locked to a single application until the payload is removed. The home button is disabled, and the device returns to the specified application automatically upon wake or reboot.

This payload is supported only in iOS 6.0 and later.

The payload contains the following key:

Key	Type	Value
App	Dictionary	A dictionary containing information about the app.

The `App` dictionary, in turn, contains the following key:

Key	Type	Value
Identifier	String	The bundle identifier of the application.
Options	Dictionary	Optional. Described below. **Availability:** Available only in iOS 7.0 and later.
UserEnabledOptions	Dictionary	Optional. Described below. **Availability:** Available only in iOS 7.0 and later.

The `Options` dictionary, if present, can contain the following keys (in iOS 7.0 and later):

Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
DisableTouch	String	Optional. If true, the touch screen is disabled. Default is false.
DisableDeviceRotation	String	Optional. If true, device rotation sensing is disabled. Default is false.
DisableVolumeButtons	String	Optional. If true, the volume buttons are disabled. Default to false.
DisableRingerSwitch	String	Optional. If true, the ringer switch is disabled. Default is false. When disabled, the ringer behavior depends on what position the switch was in when it was first disabled.
DisableSleepWakeButton	String	Optional. If true, the sleep/wake button is disabled. Default is false.
DisableAutoLock	String	Optional. If true, the device will not automatically go to sleep after an idle period.
EnableVoiceOver	String	Optional. If true, VoiceOver is turned on. Default is false.
EnableZoom	String	Optional. If true, Zoom is turned on. Default is false.
EnableInvertColors	String	Optional. If true, Invert Colors is turned on. Default is false.
EnableAssistiveTouch	String	Optional. If true, AssistiveTouch is turned on. Default is false.
EnableSpeakSelection	String	Optional. If true, Speak Selection is turned on. Default is false.
EnableMonoAudio	String	Optional. If true, Mono Audio is turned on. Default is false.

The UserEnabledOptions dictionary, if present, can contain the following keys (in iOS 7.0 and later):

Key	Type	Value
VoiceOver	String	Optional. If true, allow VoiceOver adjustment. Default is false.
Zoom	String	Optional. If true, allow Zoom adjustment. Default is false.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`InvertColors`	String	Optional. If `true`, allow Invert Colors adjustment. Default is `false`.
`AssistiveTouch`	String	Optional. If `true`, allow AssistiveTouch adjustment. Default is `false`.

CalDAV Payload

This payload configures a CalDAV account.

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

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

Key	Type	Value
`CalDAVAccountDescription`	String	Optional. The description of the account.
`CalDAVHostName`	String	The server address. In OS X, this key is required.
`CalDAVUsername`	String	The user's login name. In OS X, this key is required.
`CalDAVPassword`	String	Optional. The user's password
`CalDAVUseSSL`	Boolean	Whether or not to use SSL. In OS X, this key is optional.
`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 calendar subscription payload is designated by specifying `com.apple.subscribedcalendar.account` as the `PayloadType` value.

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

The calendar subscription payload is not supported in OS X.

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

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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	Boolean	Whether or not to use SSL.

CardDAV Payload

The CardDAV payload is designated by specifying `com.apple.carddav.account` as the `PayloadType` value.

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

This payload is only available in OS X v10.8 and later. As of OS X v10.8 and later, this payload type supports obtaining "CardDAVUsername" and "CardDAVPassword" from an Identification Payload, if present.

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

Key	Type	Value
CardDAVAccountDescription	String	Optional. The description of the account.
CardDAVHostName	String	The server address.
CardDAVUsername	String	The user's login name.
CardDAVPassword	String	Optional. The user's password
CardDAVUseSSL	Boolean	Optional. Whether or not to use SSL.
CardDAVPort	Number	Optional. The port on which to connect to the server.
CalDAVPrincipalURL	String	Optional. The base URL to the user's calendar.

Cellular Payload

A cellular payload configures cellular network settings on the device. In iOS 7 and later, a cellular payload is designated by specifying `com.apple.cellular` as the `PayloadType` value. Cellular payloads have two important installation requirements:

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

- No more than one cellular payload can be installed at any time.
- A cellular payload cannot be installed if an APN payload is already installed.

This payload replaces the `com.apple.managedCarrier` payload, which is supported, but deprecated.

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

Key	Type	Value
`AttachAPN`	Dictionary	Optional. An `AttachAPN` configuration dictionary, described below.
`APNs`	Array	Optional. An array of APN dictionaries, described below. Only the first entry is currently used.

The `AttachAPN` dictionary contains the following keys:

Key	Type	Value
`Name`	String	Required. A name for this configuration.
`AuthenticationType`	String	Optional. Must contain either `CHAP` or `PAP`. Defaults to `PAP`.
`Username`	String	Optional. A user name used for authentication.
`Password`	String	Optional. A password used for authentication.

Each `APN` dictionary contains the following keys:

Key	Type	Value
`Name`	String	Required. A name for this configuration.
`AuthenticationType`	String	Optional. Must contain either `CHAP` or `PAP`. Defaults to `PAP`.
`Username`	String	Optional. A user name used for authentication.
`Password`	String	Optional. A password used for authentication.
`ProxyServer`	String	Optional. The proxy server's network address.
`ProxyServerPort`	String	Optional. The proxy server's port.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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`.
`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`	Boolean	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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`OutgoingPassword`	String	Optional. Password for the Outgoing Mail Server. Use only with encrypted profiles.
`OutgoingPasswordSame-AsIncomingPassword`	Boolean	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`	Boolean	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.
`PreventMove`	Boolean	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 only in iOS 5.0 and later.
`PreventAppSheet`	Boolean	Optional. Default `false`. If `true`, this account is not available for sending mail in third-party applications. **Availability:** Available only in iOS 5.0 and later.
`SMIMEEnabled`	Boolean	Optional. Default `false`. If `true`, this account supports S/MIME. **Availability:** Available only in iOS 5.0 and later.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
SMIMESigning-CertificateUUID	String	Optional. The PayloadUUID of the identity certificate used to sign messages sent from this account. **Availability:** Available only 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 only in iOS 5.0 and later.
disableMailRecents-Syncing	Boolean	If true, this account is excluded from address Recents syncing. This defaults to false. **Availability:** Available only in iOS 6.0 and later.

Exchange Payload

In iOS, the Exchange payload is designated by specifying com.apple.eas.account as the PayloadType value. This payload configures an Exchange Active Sync account on the device.

In OS X, the Exchange payload is designated by specifying com.apple.ews.account as the PayloadType value. This payload will configure an Exchange Web Services account. OS X will only configure an Exchange account for Contacts. Mail and Calendar are not configured using this profile.

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. In OS X, this key is required.
Host	String	Specifies the Exchange server host name (or IP address). In OS X, this key is required.
SSL	Boolean	Optional. Default YES. Specifies whether the Exchange server uses SSL for authentication.
UserName	String	This string specifies the user name for this Exchange account. If missing, the devices prompts for it during profile installation. In OS X, this key is required.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`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`	Boolean	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`	Boolean	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`	Boolean	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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
disableMailRecents-Syncing	Boolean	If true, this account is excluded from address Recents syncing. This defaults to false. **Availability:** Available only in iOS 6.0 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.

FileVault Recovery Key Redirection Payload

FileVault full-disk encryption (FDE) recovery keys are, by default, sent to Apple if the user requests it. With this key, you can redirect those recovery keys to a corporate server. Only one payload of this type is allowed per system.

com.apple.security.FDERecoveryRedirect

Key	Type	Value
RedirectURL	String	The URL to which FDE recovery keys should be sent instead of Apple. Must begin with https://.
EncryptCertPayload-UUID	String	The UUID of a payload within the same profile that contains a certificate whose public key is used to encrypt the recovery key when it is sent to the redirected URL. The referenced payload must be of type com.apple.security.pkcs1.

Once installed, this payload causes any FileVault recovery keys to be redirected to the specified URL instead of being sent to Apple. This will require sites to implement their own HTTPS server that will receive the recovery keys via a POST request. Details of the data sent to the server will be provided in a different document.

This payload is valid only in system-scoped profiles (where PayloadScope is System).

Font Payload

A Font payload lets you add an additional font to an iOS device. Font payloads are designated by specifying com.apple.font as the PayloadType value. You can include multiple Font payloads, as needed.

This payload is supported only in iOS 7.0 and later.

A Font payload contains the following keys:

Key	Type	Value
Name	String	Optional. The user-visible name for the font. This field is replaced by the actual name of the font after installation.
Font	Data	The contents of the font file.

Each payload must contain exactly one font file in TrueType (.ttf) or OpenType (.otf) format. Collection formats (.ttc or .otc) are not supported.

Important: Fonts are identified by their embedded PostScript names. Two fonts with the same PostScript name are considered to be the same font even if their contents differ. Installing two different fonts with the same PostScript name is not supported, and the resulting behavior is undefined.

Global HTTP Proxy Payload

The Global HTTP Proxy payload is designated by specifying `com.apple.proxy.http.global` as the `PayloadType`.

This payload allows you to specify global HTTP proxy settings.

There can only be one of this payload at any time. This payload can only be installed on a supervised device.

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

Key	Type	Value
ProxyType	String	If you choose manual proxy type, you need the proxy server address including its port and optionally a username and password into the proxy server. If you choose auto proxy type, you can enter a proxy autoconfiguration (PAC) URL.
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.
ProxyPassword	String	Optional. The password used to authenticate to the proxy server.
ProxyPACURL	String	Optional. The URL of the PAC file that defines the proxy configuration.

Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
ProxyPACFallback-Allowed	Boolean	Optional. If false, prevents the device from connecting directly to the destination if the PAC file is unreachable. Default is true. **Availability:** Available in iOS 7 and later.
ProxyCaptiveLogin-Allowed	Boolean	Optional. If true, allows the device to bypass the proxy server to display the login page for captive networks. Default is false. **Availability:** Available in iOS 7 and later.

If the ProxyType field is set to Auto and no ProxyPACURL value is specified, the device uses the web proxy autodiscovery protocol (WPAD) to discover proxies.

Identification Payload

The Removal Password payload is designated by specifying com.apple.configurationprofile.identification value as the PayloadType value.

This payload allows you to save names of the account user and prompt text. If left blank, the user has to provide this information when he or she installs the profile.

The Identification payload is not supported in iOS.

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

Key	Type	Value
FullName	String	The full name of the designated accounts.
EmailAddress	String	The address for the accounts.
UserName	String	The UNIX user name for the accounts.
Password	String	You can provide the password or choose to have the user provide it when he or she installs the profile.
Prompt	String	Custom instruction for the user, if needed.

LDAP Payload

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

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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`	Boolean	Whether or not to use SSL.
`LDAPAccountUserName`	String	Optional. The username.
`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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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`	Boolean	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`	Boolean	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. In OS X, this will be translated to screensaver settings.
`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 &%$#.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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`	Boolean	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. **Availability:** This key is unavailable in OS X.
`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. In OS X, this will be translated to screensaver settings.

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.

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.

The Restrictions payload is not supported in OS X.

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

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`allowAccount-Modification`	Boolean	Optional. Supervised only. If set to `false`, account modification is disabled. **Availability:** Available only in iOS 7.0 and later.
`allowAirDrop`	Boolean	Optional. Supervised only. If set to `false`, AirDrop is disabled. **Availability:** Available only in iOS 7.0 and later.
`allowAppCellularData-Modification`	Boolean	Optional. Supervised only. If set to `false`, changes to cellular data usage for apps are disabled. **Availability:** Available only in iOS 7.0 and later.
`allowAppInstallation`	Boolean	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`	Boolean	Optional. When `false`, disables Siri. Defaults to `true`.
`allowAssistantUser-GeneratedContent`	Boolean	Optional. Supervised only. When `false`, prevents Siri from querying user-generated content from the web. **Availability:** Available in iOS 7 and later.
`allowAssistant-WhileLocked`	Boolean	Optional. When `false`, the user is unable to use Siri when the device is locked. Defaults to `true`. This restriction is ignored if the device does not have a passcode set. **Availability:** Available only in iOS 5.1 and later.
`allowBookstore`	Boolean	Optional. Supervised only. If set to `false`, iBookstore will be disabled. This will default to `true`. **Availability:** Available in iOS 6.0 and later.
`allowBookstore-Erotica`	Boolean	Optional. Supervised only prior to iOS 6.1. If set to `false`, the user will not be able to download media from the iBookstore that has been tagged as erotica. This will default to `true`. **Availability:** Available in iOS 6.0 and later.
`allowCamera`	Boolean	Optional. When `false`, the camera is completely disabled and its icon is removed from the Home screen. Users are unable to take photographs.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`allowCloudBackup`	Boolean	Optional. When `false`, disables backing up the device to iCloud. **Availability:** Available in iOS 5.0 and later.
`allowCloudDocument-Sync`	Boolean	Optional. When `false`, disables document and key-value syncing to iCloud. **Availability:** Available in iOS 5.0 and later.
`allowCloudKeychain-Sync`	Boolean	Optional. If `false`, disables keychain syncing to iCloud. Default is `true`. **Availability:** Available only in iOS 7.0 and later.
`allowDiagnostic-Submission`	Boolean	Optional. When `false`, this prevents the device from automatically submitting diagnostic reports to Apple. Defaults to `true`. **Availability:** Available only in iOS 6.0 and later.
`allowExplicitContent`	Boolean	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.
`allowFindMyFriends-Modification`	Boolean	Optional. Supervised only. If set to `false`, changes to Find My Friends are disabled. **Availability:** Available only in iOS 7.0 and later.
`allowFingerprint-ForUnlock`	Boolean	Optional. If `false`, prevents Touch ID from unlocking a device. **Availability:** Available in iOS 7 and later.
`allowGameCenter`	Boolean	Optional. Supervised only. When `false`, Game Center is disabled and its icon is removed from the Home screen. Default is `true`. **Availability:** Available only in iOS 6.0 and later.
`allowHostPairing`	Boolean	Supervised only. If set to `false`, host pairing is disabled with the exception of the supervision host. If no supervision host certificate has been configured, all pairing is disabled. **Availability:** Available only in iOS 7.0 and later.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`allowLockScreenControlCenter`	Boolean	Optional. If `false`, prevents Control Center from appearing on the Lock screen. **Availability:** Available in iOS 7 and later.
`allowLockScreenNotificationsView`	Boolean	Optional. If set to `false`, the Notifications view in Notification Center on the lock screen is disabled. **Availability:** Available only in iOS 7.0 and later.
`allowLockScreenTodayView`	Boolean	Optional. If set to `false`, the Today view in Notification Center on the lock screen is disabled. **Availability:** Available only in iOS 7.0 and later.
`allowOpenFromManagedToUnmanaged`	Boolean	Optional. If `false`, documents in managed apps and accounts only open in other managed apps and accounts. Default is `true`. **Availability:** Available only in iOS 7.0 and later.
`allowOpenFromUnmanagedToManaged`	Boolean	Optional. If set to `false`, documents in unmanaged apps and accounts will only open in other unmanaged apps and accounts. Default is `true`. **Availability:** Available only in iOS 7.0 and later.
`allowOTAPKIUpdates`	Boolean	Optional. If `false`, over-the-air PKI updates are disabled. Default is `true`. **Availability:** Available only in iOS 7.0 and later.
`allowPassbookWhileLocked`	Boolean	Optional. If set to `false`, Passbook notifications will not be shown on the lock screen.This will default to `true`. **Availability:** Available in iOS 6.0 and later.
`allowPhotoStream`	Boolean	Optional. When `false`, disables Photo Stream. **Availability:** Available in iOS 5.0 and later.
`allowSafari`	Boolean	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.
`allowScreenShot`	Boolean	Optional. When `false`, users are unable to save a screenshot of the display.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`allowSharedStream`	Boolean	Optional. If set to `false`, Shared Photo Stream will be disabled. This will default to `true`. **Availability:** Available in iOS 6.0 and later.
`allowUIConfiguration-ProfileInstallation`	Boolean	Optional. Supervised only. If set to `false`, the user is prohibited from installing configuration profiles and certificates interactively. This will default to `true`. **Availability:** Available in iOS 6.0 and later.
`allowUntrusted-TLSPrompt`	Boolean	Optional. When `false`, automatically rejects untrusted HTTPS certificates without prompting the user. **Availability:** Available in iOS 5.0 and later.
`allowYouTube`	Boolean	Optional. When `false`, the YouTube application is disabled and its icon is removed from the Home screen. This key is ignored in iOS 6 and later because the YouTube app is not provided.
`allowiTunes`	Boolean	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.
`autonomousSingleApp-ModePermittedAppIDs`	Array of strings	Optional. Supervised only. If present, allows apps identified by the bundle IDs listed in the array to autonomously enter Single App Mode. **Availability:** Available only in iOS 7.0 and later.
`forceITunesStore-PasswordEntry`	Boolean	Optional. When `true`, forces user to enter their iTunes password for each transaction. **Availability:** Available in iOS 5.0 and later.
`forceLimitAdTracking`	Boolean	Optional. If `true`, limits ad tracking. Default is `false`. **Availability:** Available only in iOS 7.0 and later.

SCEP Payload

The SCEP (Simple Certificate Enrollment Protocol) payload is designated by specifying `com.apple.security.scep` as the `PayloadType` value.

An SCEP payload automates the request of a client certificate from an SCEP server, as described in *Over-the-Air Profile Delivery and Configuration*.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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.
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 certificate authorities, such as Windows CA, support only encryption or signing, but not both at the same time. **Availability:** Available only in iOS 4 and later.

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 50) or read *Over-the-Air Profile Delivery and Configuration* .

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

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* .

Single Sign-On Account Payload

The Single Sign-On Account payload is designated by specifying `com.apple.sso` as the `PayloadType`.

This payload is supported only in iOS 7.0 and later.

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

Key	Type	Value
`Name`	String	Human-readable name for the account.
`Kerberos`	Dictionary	Kerberos-related information, described below.

The Kerberos `dictionary` can contain the following keys:

Key	Type	Value
`PrincipalName`	String	Optional. The Kerberos principal name. If not provided, the user is prompted for one during profile installation. This field must be provided for MDM installation.
`Realm`	String	The Kerberos realm name. This value should be properly capitalized.
`URLPrefixMatches`	Array of strings	List of URLs prefixes that must be matched in order to use this account for Kerberos authentication over HTTP.
`AppIdentifierMatches`	Array of strings	Optional. List of app identifiers that are allowed to use this login. If this field missing, this login matches all app identifiers. This array, if present, may not be empty.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Each entry in the `URLPrefixMatches` array must contain a URL prefix. Only URLs that begin with one of the strings in this account are allowed to access the Kerberos ticket. URL matching patterns must include the scheme—for example, `http://www.example.com/`. If a matching pattern does not end in `/`, a `/` is appended to it.

The URL matching patterns must begin with either `http://` or `https://`. A simple string match is performed, so the URL prefix `http://www.example.com/` does not match `http://www.example.com:80/`.

The pattern `http://` and `https://` matches all HTTP and HTTPS URLs, respectively.

The `AppIdentifierMatches` array must contain strings that match app bundle IDs. These strings may be exact matches (`com.mycompany.myapp`, for example) or may specify a prefix match on the bundle ID by using the `*` wildcard character. The wildcard character must appear after a period character (`.`), and may appear only at the end of the string (`com.mycompany.*`, for example). When a wildcard is given, any app whose bundle ID begins with the prefix is granted access to the account.

System Policy Control Payload

The System Policy Control payload is designated by specifying `com.apple.systempolicy.control` as the `PayloadType`. This is one of three payloads that allows control of various GateKeeper settings.

This payload allows control over configuring the “Allowed applications downloaded from:” option in the “General” tab of “Security & Privacy” in System Preferences.

This payload must only exist in a device profile. If the payload is present in a user profile, an error will be generated during installation and the profile will fail to install.

This payload is supported only on OS X v10.8 and later.

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

Key	Type	Value
`EnableAssessment`	Boolean	Optional. If the key is present and has a value of `YES`, Gatekeeper is enabled. If the key is present and has a value of `NO`, Gatekeeper is disabled.
`AllowIdentified-Developers`	Boolean	Optional. If the key is present and has a value of `YES`, Gatekeeper’s “Mac App Store and identified developers” option is chosen. If the key is present and has a value of `NO`, Gatekeeper’s “Mac App Store” option is chosen. If `EnableAssessment` is not `true`, this key has no effect.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

System Policy Rule Payload

The System Policy Control payload is designated by specifying `com.apple.systempolicy.control` as the `PayloadType`. This is one of three payloads that allows control of various GateKeeper settings.

This payload allows control over Gatekeeper's system policy rules. The keys and functionality are tightly related to the `spctl` command line tool. You should be read the manual page for `spctl`.

This payload must only exist in a device profile. If the payload is present in a user profile, an error will be generated during installation and the profile will fail to install.

This payload is supported only on OS X v10.8 and later.

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

Key	Type	Value
`Requirement`	String	The policy requirement. This key must follow the syntax described in "Code Signing Requirement Language".
`Comment`	String	Optional. This string will appear in the System Policy UI. If it is missing, "PayloadDisplayName" or "PayloadDescription" will be put into this field before the rule is added to the System Policy database.
`Expiration`	Date	Optional. An expiration date for rule(s) being processed.
`OperationType`	String	Optional. One of `operation:execute`, `operation:install`, or `operation:lsopen`. This will default to `operation:execute`.

The client has no way to display information about what certificate is being accepted by the signing requirement if the requirement keys is specified as:

```
certificate leaf = H"7696f2cbf7f7d43fceb879f52f3cdc8fadfccbd4"
```

You can embed the certificate within the payload itself, allowing the Profiles preference pane and System Profile report to display information about the certificate(s) being used. To do so, specify the `Requirement` key using a payload variable of the form $HASHCERT_xx$ where "xx" is the name of an additional key within the same payload that contains the certificate data in DER format.

For example, if you specify:

```
<key>Requirement</key>
<string>certificate leaf = $HASHCERT_Cert1Data$</string>
```

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

and then provide:

```
<key>Cert1Data</key>
<data>
MIIFTDCCBDSgAwIBAgIHBHXzxGzq8DANBgkqhkiG9w0BAQUFADCByjELMAkGA1UEBhMC
...
z1I6yBET5qaGhpWexEp3baLbXLcrtgufmDSUtUnImavGyw==
</data>
```

The client will get the value of `Cert1Data` key, perform a SHA1 hash on it and use the resulting requirement string of:

```
certificate leaf = H"7696f2cbf7f7d43fceb879f52f3cdc8fadfccbd4"
```

If you want, you may reference multiple $HASHCERT_xx$ within the requirement string.

System Policy Managed Payload

The System Policy Managed payload is designated by specifying `com.apple.systempolicy.managed` as the `PayloadType`. This is one of three payloads that allows control of various GateKeeper settings.

This payload allows control to disable the Finder's contextual menu that allows bypass of System Policy restrictions.

This payload is supported only on OS X v10.8 and later.

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

Key	Type	Value
`DisableOverride`	Boolean	Optional. If YES, the Finder's contextual menu item will be disabled.

VPN Payload

The VPN payload is used for traditional systemwide VPNs based on L2TP, PPTP, and IPSec. This payload should not be confused with the Per-App VPN, described in "Per-App VPN Payload" (page 11).

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:

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`UserDefinedName`	String	Description of the VPN connection displayed on the device.
`OverridePrimary`	Boolean	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. This key can also be “VPN” to support additional services via it’s VPNSubType key.
`OnDemandEnabled`	Integer	`1` if the VPN connection should be brought up on demand, else `0`.
`OnDemandMatchDomains-Always`	Array of Strings	*Deprecated.* A list of domain names. In versions of iOS prior to iOS 7, if the hostname ends with one of these domain names, the VPN is started automatically. In iOS 7 and later, if this key is present, the associated domain names are treated as though they were associated with the `OnDemandMatchDomainsOnRetry` key. This behavior can be overridden by `OnDemandRules`.
`OnDemandMatch-DomainsNever`	Array of Strings	*Deprecated.* A list of domain names. If the hostname ends with one of these domain names, the VPN is *not* started automatically. This might be used to exclude a subdomain within an included domain. This behavior can be overridden by `OnDemandRules`. In iOS 7 and later, this key is deprecated (but still supported) in favor of `EvaluateConnection` actions in the `OnDemandRules` dictionaries.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`OnDemandMatchDomains–OnRetry`	Array of Strings	*Deprecated.* A list of domain names. If the hostname ends with one of these domain names, if a DNS query for that domain name fails, the VPN is started automatically. This behavior can be overridden by `OnDemandRules`. In iOS 7 and later, this key is deprecated (but still supported) in favor of `EvaluateConnection` actions in the `OnDemandRules` dictionaries.
`OnDemandRules`	Array of Dictionaries	Determines when and how an on-demand VPN should be used. See "On Demand Rules Dictionary Keys" (page 40) for details.
`VendorConfig`	Dictionary	A dictionary for configuration information specific to a given third-party VPN solution.

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.
`CCPMPPE40Enabled`	Boolean	See discussion under `CCPEnabled`. Used for PPTP.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
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 is 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 is SharedSecret. The value is KeyID. Used for L2TP and Cisco IPSec.
SharedSecret	Data	The shared secret for this VPN account. Only present if AuthenticationMethod is 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 is Certificate. Used for Cisco IPSec.
PromptForVPNPIN	Boolean	Tells whether to prompt for a PIN when connecting. Used for Cisco IPSec.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

On Demand Rules Dictionary Keys

The `OnDemandRules` key in a VPN payload is associated with an array of dictionaries that define the network match criteria that identify a particular network location.

In typical use, VPN On Demand matches the dictionaries in the `OnDemandRules` array against properties of your current network connection to determine whether domain-based rules should be used in determining whether to connect, then handles the connection as follows:

- If domain-based matching is enabled for a matching `OnDemandRules` dictionary, then for each dictionary in that dictionary's `EvaluateConnection` array, VPN On Demand compares the requested domain against the domains listed in the `Domains` array.
- If domain-based matching is not enabled, the specified behavior (usually `Connect`, `Disconnect`, or `Ignore`) is used if the dictionary otherwise matches.

> **Note:** For backwards compatibility, VPN On Demand also allows you to specify the `Allow` action, in which case the domains to match are determined by arrays in the VPN payload itself (`OnDemandMatchDomainsAlways`, `OnDemandMatchDomainsOnRetry`, and `OnDemandMatchDomainsNever`). However, this is deprecated in iOS 7.

Whenever a network change is detected, the VPN On Demand service compares the newly connected network against the match network criteria specified in each dictionary (in order) to determine whether VPN On Demand should be allowed or not on the newly joined network. The matching criteria can include any of the following:

- DNS domain or DNS server settings (with wildcard matching)
- SSID
- Interface type
- reachable server detection

Dictionaries are checked sequentially, beginning with the first dictionary in the array. A dictionary matches the current network only if *all* of the specified policies in that dictionary match. You should always set a default behavior for unknown networks by specifying an action with no matching criteria as the last dictionary in the array.

If a dictionary matches the current network, a server probe is sent if a URL is specified in the profile. VPN then acts according to the policy defined in the dictionary (for example, allow VPNOnDemand, ignore VPNOnDemand, connect, or disconnect).

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Important: Be sure to set a catch-all value. If you do not, the current default behavior is to allow the connection to occur, but this behavior is not guaranteed.

The `OnDemandRules` dictionaries can contain one or more of the following keys:

Key	Type	Value
`Action`	String	The action to take if this dictionary matches the current network. Possible values are: `Allow`—*Deprecated.* Allow VPN On Demand to connect if triggered. `Connect`—Unconditionally initiate a VPN connection on the next network attempt. `Disconnect`—Tear down the VPN connection and do not reconnect on demand as long as this dictionary matches. `EvaluateConnection`—Evaluate the `ActionParameters` array for each connection attempt. `Ignore`—Leave any existing VPN connection up, but do not reconnect on demand as long as this dictionary matches.
`ActionParameters`	Array of dictionaries	A dictionary that provides rules similar to the `OnDemandRules` dictionary, but evaluated on each connection instead of when the network changes. These dictionaries are evaluated in order, and the behavior is determined by the first dictionary that matches. The keys allowed in each dictionary are described in Table 1-1 (page 42). **Note:** This array is used only for dictionaries in which `EvaluateConnection` is the `Action` value.

Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`DNSDomainMatch`	Array of strings	An array of domain names. This rule matches if any of the domain names in the specified list matches any domain in the device's search domains list. A wildcard '*' prefix is supported. For example, `*.example.com` matches against either `mydomain.example.com` or `yourdomain.example.com`.
`DNSServerAddress-Match`	Array of strings	An array of IP addresses. This rule matches if any of the network's specified DNS servers match any entry in the array. Matching with a single wildcard is supported. For example, `17.*` matches any DNS server in the class A 17 subnet.
`InterfaceTypeMatch`	String	An interface type. If specified, this rule matches only if the primary network interface hardware matches the specified type. Supported values are `Ethernet`, `WiFi`, and `Cellular`.
`SSIDMatch`	String	An array of SSIDs to match against the current network. If the network is not a Wi-Fi network or if the SSID does not appear in this array, the match fails. Omit this key and the corresponding array to match against any SSID.
`URLStringProbe`	String	A URL to probe. If this URL is successfully fetched (returning a `200` HTTP status code) without redirection, this rule matches.

The keys allowed in each `ActionParameters` dictionary are described in Table 1-1.

Table 1-1 Keys in the `ActionParameters` dictionary

Key	Type	Value
`Domains`	Array of strings	*Required.* The domains for which this evaluation applies.

Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
DomainAction	String	*Required.* Defines the VPN behavior for the specified domains. Allowed values are: ConnectIfNeeded—The specified domains should trigger a VPN connection attempt if domain name resolution fails, such as when the DNS server indicates that it cannot resolve the domain, responds with a redirection to a different server, or fails to respond (timeout). NeverConnect—The specified domains should never trigger a VPN connection attempt.
RequiredDNSServers	Array of strings	*Optional.* An array of IP addresses of DNS servers to be used for resolving the specified domains. These servers need not be part of the device's current network configuration. If these DNS servers are not reachable, a VPN connection is established in response. These DNS servers should be either internal DNS servers or trusted external DNS servers. **Note:** This key is valid only if the value of DomainAction is ConnectIfNeeded.
RequiredURLString-Probe	String	*Optional.* An HTTP or HTTPS (preferred) URL to probe, using a GET request. If the URL's hostname cannot be resolved, if the server is unreachable, or if the server does not respond with a 200 HTTP status code, a VPN connection is established in response. **Note:** This key is valid only if the value of DomainAction is ConnectIfNeeded.

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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`Label`	String	The name of the Web Clip as displayed on the Home screen.
`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`	Boolean	Optional. If No, the user cannot remove the Web Clip, but it will be removed if the profile is deleted.

Web Content Filter Payload

The Web Content Filter payload allows you to whitelist and blacklist specific web URLs. It is designated by specifying `com.apple.webcontent-filter` as the `PayloadType` value.

This payload is supported only in iOS 7.0 and later, and only on supervised devices.

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

Key	Type	Value
`AutoFilterEnabled`	Boolean	Optional. If `true`, automatic filtering will be enabled. Defaults to `false`.
`PermittedURLs`	Array of strings	Optional. Used only when `AutoFilterEnabled` is `true`. Otherwise, this field is ignored. Each entry contains a URL that is accessible whether the automatic filter allows access or not.
`WhitelistedBookmarks`	Array of dictionaries	Optional. If present, these URLs are added to the browser's bookmarks, and the user is not allowed to visit any sites other than these.
`BlacklistedURLs`	Array of strings	Optional. Access to the specified URLs is blocked.

Each entry in the `WhitelistedBookmarks` field contains a dictionary with the following keys:

Key	Type	Value
`URL`	String	URL of the whitelisted bookmark.
`BookmarkPath`	String	Optional. The folder into which the bookmark should be added in Safari—`/Interesting Topic Pages/Biology/`, for example. If absent, the bookmark is added to the default bookmarks directory.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
Title	String	The title of the bookmark.

When multiple content filter payloads are present:

- The blacklist is the union of all blacklists—that is, any URL that appears in any blacklist is inaccessible.
- The permitted list is the intersection of all permitted lists—that is, only URLs that appear in *every* permitted list are accessible when they would otherwise be blocked by the automatic filter.
- The whitelist list is the intersection of all whitelists—that is, only URLs that appear in *every* whitelist are accessible.

URLs are matched using string-based prefix matching—that is, a URL matches a whitelist, blacklist, or permitted list pattern if the exact characters of the pattern appear at the beginning of the URL. No attempt is made to match aliases (IP address versus DNS names, for example) or to handle requests with explicit port numbers. URLs should begin with `http://` or `https://`. If necessary, separate entries must be made for `http://` and `https://` versions of the same URL.

If a profile does not contain an array for `PermittedURLs` or `WhitelistedBookmarks`, that profile is skipped when evaluating the missing array or arrays. As an exception, if a payload contains an `AutoFilterEnabled` key, but does not contain a `PermittedURLs` array, that profile is treated as containing an empty array—that is, all websites are blocked.

All filtering options are active simultaneously. Only URLs and sites that pass *all* rules are permitted.

Wi-Fi Payload

The Wi-Fi payload is designated by specifying `com.apple.wifi.managed` as the `PayloadType` 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. In iOS 7.0 and later, this is optional if a `DomainName` value is provided
HIDDEN_NETWORK	Boolean	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`.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
AutoJoin	Boolean	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.
IsHotspot	Boolean	Optional. Default false. If true, the network is treated as a hotspot. **Availability:** Available in iOS 7.0 and later.
DomainName	String	Optional. Domain Name used for Wi-Fi Hotspot 2.0 negotiation. This field can be provided instead of SSID_STR. **Availability:** Available in iOS 7.0 and later.
ServiceProvider-RoamingEnabled	Boolean	Optional. If true, allows connection to roaming service providers. **Availability:** Available in iOS 7.0 and later.
RoamingConsortiumOIs	Array of strings	Optional. Array of Roaming Consortium Organization Identifiers used for Wi-Fi Hotspot 2.0 negotiation. **Availability:** Available in iOS 7.0 and later.
NAIRealmNames	Array of strings	Optional. Array of strings. List of Network Access Identifier Realm names used for Wi-Fi Hotspot 2.0 negotiation. **Availability:** Available in iOS 7.0 and later.
MCCAndMNCs	Array of strings	Optional. Array of strings. List of Mobile Country Code (MCC)/Mobile Network Code (MNC) pairs used for Wi-Fi Hotspot 2.0 negotiation. Each string must contain exactly six digits. **Availability:** Available in iOS 7.0 and later.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
DisplayedOperator-Name	String	**Availability:** Available in iOS 7.0 and later.
ProxyType	String	Optional. Valid values are None, Manual, and Auto. **Availability:** Available in iOS 5.0 and later.

If the EncryptionType field is set to WEP, WPA, or ANY, the following fields may also be provided:

Key	Type	Value
Password	String	Optional.
EAPClientConfiguration	Dictionary	Described in "EAPClientConfiguration Dictionary" (page 48).
PayloadCertificateUUID	String	Described in "Certificates" (page 50).

Note: The absence of a password does not prevent a network from being added to the list of known networks. The user is eventually prompted to provide the password when connecting to that network.

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.
ProxyPassword	String	Optional. The password used to authenticate to the proxy server.
ProxyPACURL	String	Optional. The URL of the PAC file that defines the proxy configuration.
ProxyPACFallback-Allowed	Boolean	Optional. If false, prevents the device from connecting directly to the destination if the PAC file is unreachable. Default is true. **Availability:** Available in iOS 7 and later.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

If the `ProxyType` field is set to `Auto` and no `ProxyPACURL` value is specified, the device uses the web proxy autodiscovery protocol (WPAD) to discover proxies.

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`.
`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`.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Key	Type	Value
`TLSAllowTrust-Exceptions`	Boolean	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`.
`TLSCertificateIs-Required`	`Boolean`	Optional. If `true`, allows for two-factor authentication for EAP-TTLS, PEAP, or EAP-FAST. If `false`, allows for zero-factor authentication for EAP-TLS. The default is `true` for EAP-TLS, and `false` for other EAP types. **Availability:** Available in iOS 7.0 and later.
`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 *Configuration Profile Reference*.

Date	Notes
2013-10-22	Added information about the keychain syncing restriction.
2013-10-01	Removed unsupported keys from document.
2013-09-18	Updated with a few additional iOS 7 keys.
2012-12-13	Corrected minor technical and typographical errors.
2012-09-22	Made minor typographical fixes and clarified a few details specific to OS X.
2012-09-19	Updated document for iOS 6 and added support for OS X 10.8.
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.

2013-10-22 | Copyright © 2013 Apple Inc. All Rights Reserved. Apple Confidential Information.

Apple Inc.
Copyright © 2013 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

Apple, the Apple logo, AirPlay, Bonjour, FileVault, Finder, iBook, iBooks, iPad, iPhone, iTunes, Keychain, Mac, OS X, Pages, Passbook, Safari, Siri, and TrueType are trademarks of Apple Inc., registered in the U.S. and other countries.

AirDrop and AirPrint are trademarks of Apple Inc.

iCloud, iTunes Music Store, and iTunes Store are service marks of Apple Inc., registered in the U.S. and other countries.

App Store, iBookstore, and Mac App Store are service marks of Apple Inc.

UNIX is a registered trademark of The Open Group.

Xerox is a registered trademark of Xerox Corporation.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.