

Configuration Profile Reference

Developer

Contents

Configuration Profile Keys	5
Payload Dictionary Keys Common to All Payloads	7
Payload-Specific Property Keys	7
Active Directory Certificate Profile Payload	8
AirPlay Payload	9
AirPlay Security Payload	10
AirPrint Payload	11
App Lock Payload	12
AppStore Payload	14
Autonomous Single App Mode	15
CalDAV Payload	16
Calendar Subscription Payload	16
CardDAV Payload	17
Cellular Payload	18
Certificate Payload	19
Certificate Preference Payload	20
Certificate Transparency Payload	21
Conference Room Display Payload	22
Content Caching Payload	23
Desktop Payload	27
DNS Proxy Payload	27
Dock Payload	27
Education Configuration Payload	29
Email Payload	33
802.1x Ethernet Payload	36
Exchange Payload	37
FileVault 2	41
FDE Recovery Key Escrow Payload	42
FileVault Client Request	43
FileVault Server Response	44
Firewall Payload	44
Font Payload	45
Global HTTP Proxy Payload	45
Global Preferences Payload	46
Google Account Payload	46
Home Screen Layout Payload	48
Identification Payload	49
Identity Preference Payload	49
Kernel Extension Policy	50

Configuration Profile Reference

Developer

Contents

Configuration Profile Keys	5
Payload Dictionary Keys Common to All Payloads	7
Payload-Specific Property Keys	7
Active Directory Certificate Profile Payload	8
AirPlay Payload	9
AirPlay Security Payload	10
AirPrint Payload	11
App Lock Payload	12
AppStore Payload	14
Autonomous Single App Mode	15
CalDAV Payload	16
Calendar Subscription Payload	16
CardDAV Payload	17
Cellular Payload	18
Certificate Payload	19
Certificate Preference Payload	20
Certificate Transparency Payload	21
Conference Room Display Payload	22
Content Caching Payload	23
Desktop Payload	27
DNS Proxy Payload	27
Dock Payload	27
Education Configuration Payload	29
Email Payload	33
802.1x Ethernet Payload	36
Exchange Payload	37
FileVault 2	41
FDE Recovery Key Escrow Payload	42
FileVault Client Request	43
FileVault Server Response	44
Firewall Payload	44
Font Payload	45
Global HTTP Proxy Payload	45
Global Preferences Payload	46
Google Account Payload	46
Home Screen Layout Payload	48
Identification Payload	49
Identity Preference Payload	49
Kernel Extension Policy	50

LDAP Payload	51
Login Items Payload	52
Loginwindow Payload	52
Media Management	54
Mobile Accounts Payload	56
Network Usage Rules Payload	56
Notifications Payload	57
NSExtension Management	58
Parental Controls Payload	58
Passcode Policy Payload	63
Privacy Preferences Policy Control Payload	64
Profile Removal Password Payload	66
Restrictions Payload	67
SCEP Payload	80
Screensaver	82
Setup Assistant	83
Shared Device Configuration Payload	83
ShareKit Payload	84
Single Sign-On Account Payload	85
SmartCard Settings Payload	86
Software Update	87
System Migration Payload	87
System Policy Control Payload	88
System Policy Rule Payload	88
System Policy Managed Payload	90
TV Remote Payload	90
Time Server Payload	91
VPN Payload	92
Per-App VPN Payload	104
App-to-Per-App VPN Mapping	105
Web Clip Payload	105
Web Content Filter Payload	106
Wi-Fi Payload	108
Domains Payload	115
Unmarked Email Domains	115
Managed Safari Web Domains	115
Active Directory Payload	117
Encrypted Profiles	119
Signing a Profile	119
Sample Configuration Profile	119
Revision History	121

LDAP Payload	51
Login Items Payload	52
Loginwindow Payload	52
Media Management	54
Mobile Accounts Payload	56
Network Usage Rules Payload	56
Notifications Payload	57
NSExtension Management	58
Parental Controls Payload	58
Passcode Policy Payload	63
Privacy Preferences Policy Control Payload	64
Profile Removal Password Payload	66
Restrictions Payload	67
SCEP Payload	80
Screensaver	82
Setup Assistant	83
Shared Device Configuration Payload	83
ShareKit Payload	84
Single Sign-On Account Payload	85
SmartCard Settings Payload	86
Software Update	87
System Migration Payload	87
System Policy Control Payload	88
System Policy Rule Payload	88
System Policy Managed Payload	90
TV Remote Payload	90
Time Server Payload	91
VPN Payload	92
Per-App VPN Payload	104
App-to-Per-App VPN Mapping	105
Web Clip Payload	105
Web Content Filter Payload	106
Wi-Fi Payload	108
Domains Payload	115
Unmarked Email Domains	115
Managed Safari Web Domains	115
Active Directory Payload	117
Encrypted Profiles	119
Signing a Profile	119
Sample Configuration Profile	119
Revision History	121

Note

This document was previously titled *iPhone Configuration Profile Reference*. It now supports both iOS and macOS.

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

Note

OSX versions 10.10 and later honor a `tr` value of the `Payloa` `ad` `Removal` `1D` `key` to prevent manual removal of profiles installed through an MDM server. Such profiles cannot be removed using the Profiles preference pane, nor the profiles command line tool even when run as root. Only the MDM server can remove such profiles. Profiles installed manually, with `Payloa` `ad` `Removal` `1D` `set` to `tr`, can be removed manually, but only by using administrative authority.

Configuration profiles are written in property list format, with `Da` `tr` 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 2](#), available in the App Store
- 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

Note

This document was previously titled *iPhone Configuration Profile Reference*. It now supports both iOS and macOS.

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

Note

OSX versions 10.10 and later honor a `tr` value of the `Payloa` `ad` `Removal` `1D` `key` to prevent manual removal of profiles installed through an MDM server. Such profiles cannot be removed using the Profiles preference pane, nor the profiles command line tool even when run as root. Only the MDM server can remove such profiles. Profiles installed manually, with `Payloa` `ad` `Removal` `1D` `set` to `tr`, can be removed manually, but only by using administrative authority.

Configuration profiles are written in property list format, with `Da` `tr` 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 2](#), available in the App Store
- 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

Note

Profile installation fails when the device is locked with a passcode.

Both iOS and macOS support 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 can be supervised when preparing them for deployment with Apple Configurator 2 (iOS 5 or later) or by using the Device Enrollment Program (iOS 7 or later). For information about Apple Configurator, go to the Mac App Store description at [Apple Configurator 2](#).

For general information about the Device Enrollment Program, visit Apple's [Corporate-owned deployments made simple](#) or [IT in Education](#). For details, go to [Apple Deployment Programs Help](#).

When a device is supervised, you can use configuration profiles to control many of its settings. This document describes the available keys in a profile and provides examples of the resulting XML payloads.

Note

Before you get started working with configuration profiles, you should create a skeleton 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
<code>PayloadContent</code>	Array	Optional. Array of payload dictionaries. Not present if <code>IsEncrypted</code> is <code>true</code> .
<code>PayloadDescription</code>	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.
<code>PayloadDisplayName</code>	String	Optional. A human-readable name for the profile. This value is displayed on the Detail screen. It does not have to be unique.
<code>PayloadExpirationDate</code>	Date	Optional. A date on which a profile is considered to have expired and can be updated over the air. This key is only used if the profile is delivered via over-the-air profile delivery.
<code>PayloadIdentifier</code>	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.
<code>PayloadOrganizationString</code>	String	Optional. A human-readable string containing the name of the organization that provided the profile.

Note

Profile installation fails when the device is locked with a passcode.

Both iOS and macOS support 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 can be supervised when preparing them for deployment with Apple Configurator 2 (iOS 5 or later) or by using the Device Enrollment Program (iOS 7 or later). For information about Apple Configurator, go to the Mac App Store description at [Apple Configurator 2](#).

For general information about the Device Enrollment Program, visit Apple's [Corporate-owned deployments made simple](#) or [IT in Education](#). For details, go to [Apple Deployment Programs Help](#).

When a device is supervised, you can use configuration profiles to control many of its settings. This document describes the available keys in a profile and provides examples of the resulting XML payloads.

Note

Before you get started working with configuration profiles, you should create a skeleton 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
<code>PayloadContent</code>	Array	Optional. Array of payload dictionaries. Not present if <code>IsEncrypted</code> is <code>true</code> .
<code>PayloadDescription</code>	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.
<code>PayloadDisplayName</code>	String	Optional. A human-readable name for the profile. This value is displayed on the Detail screen. It does not have to be unique.
<code>PayloadExpirationDate</code>	Date	Optional. A date on which a profile is considered to have expired and can be updated over the air. This key is only used if the profile is delivered via over-the-air profile delivery.
<code>PayloadIdentifier</code>	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.
<code>PayloadOrganizationString</code>	String	Optional. A human-readable string containing the name of the organization that provided the profile.

Key	Type	Content
<code>PayloadUUID</code>	String	A globally unique identifier for the profile. The actual content is unimportant, but it must be globally unique. In macOS, you can use <code>uuidgen</code> to generate reasonable UUIDs.
<code>PayloadRemovalDisposition</code>	String	Optional. Supervised only. If present and set to <code>true</code> , the user cannot delete the profile (unless the profile has a removal password and the user provides it).
<code>PayloadType</code>	String	The only supported value is <code>Configuration</code> .
<code>PayloadVersion</code>	Integer	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.
<code>PayloadScope</code>	String	Optional. 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 <code>System</code> and <code>User</code> , with <code>User</code> as the default value.
<code>Availability</code>	String	Available in macOS 10.7 and later.
<code>RemovalDate</code>	Date	Optional. The date on which the profile will be automatically removed.
<code>DurationUntilRemoval</code>	Date	Optional. Number of seconds until the profile is automatically removed. If the <code>RemovalDate</code> is present, whichever field yields the earliest date will be used.
<code>ConsentText</code>	Dictionary	Optional. A dictionary containing these keys and values:
		<ul style="list-style-type: none"> For each language in which a consent or license agreement is available, a key consisting of the IETF BCP 47 identifier for that language (for example, <code>en</code> or <code>ja</code>) and a value consisting of the agreement localized to that language. The agreement is displayed in a dialog to which the user must agree before installing the profile. The optional key <code>default</code> with its value consisting of the unlocalized agreement (usually in <code>en</code>). <p>The system chooses a localized version in the order of preference specified by the user (macOS) 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 <code>en</code> localization is used. If there is no <code>en</code> localization, then the first available localization is used.</p> <p>You should provide a default value if possible. No warning will be displayed if the user's locale does not match any localization in the <code>ConsentText</code> dictionary.</p>

Key	Type	Content
<code>PayloadUUID</code>	String	A globally unique identifier for the profile. The actual content is unimportant, but it must be globally unique. In macOS, you can use <code>uuidgen</code> to generate reasonable UUIDs.
<code>PayloadRemovalDisposition</code>	String	Optional. Supervised only. If present and set to <code>true</code> , the user cannot delete the profile (unless the profile has a removal password and the user provides it).
<code>PayloadType</code>	String	The only supported value is <code>Configuration</code> .
<code>PayloadVersion</code>	Integer	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.
<code>PayloadScope</code>	String	Optional. 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 <code>System</code> and <code>User</code> , with <code>User</code> as the default value.
<code>Availability</code>	String	Available in macOS 10.7 and later.
<code>RemovalDate</code>	Date	Optional. The date on which the profile will be automatically removed.
<code>DurationUntilRemoval</code>	Date	Optional. Number of seconds until the profile is automatically removed. If the <code>RemovalDate</code> is present, whichever field yields the earliest date will be used.
<code>ConsentText</code>	Dictionary	Optional. A dictionary containing these keys and values:
		<ul style="list-style-type: none"> For each language in which a consent or license agreement is available, a key consisting of the IETF BCP 47 identifier for that language (for example, <code>en</code> or <code>ja</code>) and a value consisting of the agreement localized to that language. The agreement is displayed in a dialog to which the user must agree before installing the profile. The optional key <code>default</code> with its value consisting of the unlocalized agreement (usually in <code>en</code>). <p>The system chooses a localized version in the order of preference specified by the user (macOS) 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 <code>en</code> localization is used. If there is no <code>en</code> localization, then the first available localization is used.</p> <p>You should provide a default value if possible. No warning will be displayed if the user's locale does not match any localization in the <code>ConsentText</code> dictionary.</p>

Note

Profile payload dictionary keys that are prefixed with "Payload" are reserved key names and must never be treated as managed preferences. Any other key in the payload dictionary may be considered a managed preference for that preference domain.

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

Payload Dictionary Keys Common to All Payloads

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 .
PayloadVersion	Integer	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 <code>payloadIdentifier</code> for 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 macOS, you can use <code>uuidgen</code> to generate reasonable UUIDs.
PayloadDisplayName	String	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](#)), each payload type contains keys that are specific to that payload type. The sections that follow describe those payload-specific keys.

Active Directory Certificate Profile Payload

The Active Directory Certificate Profile payload is designated by specifying `com.apple.ADCertificateManager` as the `PayloadValue`.

You can request a certificate from a Microsoft Certificate Authority (CA) using DCE/RPC and the Active Directory Certificate profile payload instructions detailed at <https://support.apple.com/kb/HT5357>.

This payload includes the following unique keys:

Key	Type	Value
AllowAllAppsAccess	Boolean	If true, apps have access to the private key.
CertServer	String	Fully qualified host name of the Active Directory issuing CA.
CertTemplate	String	Template Name as it appears in the General tab of the template's object in the Certificate Templates' Microsoft Management Console snap-in component.
CertificateAcquisitionString	String	Most commonly RPD using 'Web enrollment', H T T P.
CertificateAuthority	String	Name of the CA. This value is determined from the Common Name (CN) of the Active Directory entry: CN=<your CA name>, CN='Certification Authorities', CN='Public Key Services', CN='Services', or CN='Configuration', <your base Domain Name>.
CertificateRenewalTime	Integer	The number of days in advance of certificate expiration that the notification center will notify the user.
Description	String	User-friendly description of the certification identity.
KeyIsExtractable	Boolean	If true, the private key can be exported.
PromptForCredentials	Boolean	This key applies only to user certificates where Manual Download is the chosen method of profile delivery. If true, the user will be prompted for credentials when the profile is installed. Omit this key for computer certificates.
Keysize	Integer	Optional; defaults to 2048. The RSA key size for the Certificate Signing Request (CSR).
EnableAutoRenewal	Boolean	Optional. If set to true, the certificate obtained with this payload will attempt auto-renewal. Only applies to device Active Directory certificate payloads.
		Availability: Available in macOS 10.11 and later.

Active Directory Certificate Profile Payload

The Active Directory Certificate Profile payload is designated by specifying `com.apple.ADCertificateManager` as the `PayloadValue`.

You can request a certificate from a Microsoft Certificate Authority (CA) using DCE/RPC and the Active Directory Certificate profile payload instructions detailed at <https://support.apple.com/kb/HT5357>.

This payload includes the following unique keys:

Key	Type	Value
AllowAllAppsAccess	Boolean	If true, apps have access to the private key.
CertServer	String	Fully qualified host name of the Active Directory issuing CA.
CertTemplate	String	Template Name as it appears in the General tab of the template's object in the Certificate Templates' Microsoft Management Console snap-in component.
CertificateAcquisitionString	String	Most commonly RPD using 'Web enrollment', H T T P.
CertificateAuthority	String	Name of the CA. This value is determined from the Common Name (CN) of the Active Directory entry: CN=<your CA name>, CN='Certification Authorities', CN='Public Key Services', CN='Services', or CN='Configuration', <your base Domain Name>.
CertificateRenewalTime	Integer	The number of days in advance of certificate expiration that the notification center will notify the user.
Description	String	User-friendly description of the certification identity.
KeyIsExtractable	Boolean	If true, the private key can be exported.
PromptForCredentials	Boolean	This key applies only to user certificates where Manual Download is the chosen method of profile delivery. If true, the user will be prompted for credentials when the profile is installed. Omit this key for computer certificates.
Keysize	Integer	Optional; defaults to 2048. The RSA key size for the Certificate Signing Request (CSR).
EnableAutoRenewal	Boolean	Optional. If set to true, the certificate obtained with this payload will attempt auto-renewal. Only applies to device Active Directory certificate payloads.
		Availability: Available in macOS 10.13.4 and later.

AirPlay Payload

The AirPlay payload is designated by specifying `com.apple.airstplay payload`.

This payload is supported on iOS 7.0 and later and on macOS 10.10 and later.

Key	Type	Value
<code>Whitelisted</code> 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.	
<code>Passwords</code> Array of Dictionaries	Optional. If present, sets passwords for known AirPlay destinations. The dictionary format is described below.	

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

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

Each entry in the `Passwords` array is a dictionary that contains the following fields:

Key	Type	Value
<code>DeviceName</code>	The name of the AirPlay destination (used on iOS).	
<code>DeviceID</code>	The Device ID of the AirPlay destination (used on macOS).	
<code>Password</code>	The password for the AirPlay destination.	

AirPlay Payload

The AirPlay payload is designated by specifying `com.apple.airstplay payload`.

This payload is supported on iOS 7.0 and later and on macOS 10.10 and later.

Key	Type	Value
<code>Whitelisted</code> 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.	
<code>Passwords</code> Array of Dictionaries	Optional. If present, sets passwords for known AirPlay destinations. The dictionary format is described below.	

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

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

Each entry in the `Passwords` array is a dictionary that contains the following fields:

Key	Type	Value
<code>DeviceName</code>	The name of the AirPlay destination (used on iOS).	
<code>DeviceID</code>	The Device ID of the AirPlay destination (used on macOS).	
<code>Password</code>	The password for the AirPlay destination.	

AirPlay Security Payload

The AirPlay Security payload locks the Apple TV to a particular style of AirPlay Security. The AirPlay Security payload is designated by specifying `com.apple.airplaySecurityPayload`.

This payload is supported on tvOS 11.0 and later.

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

Key	Type	Value
<code>SecurityType</code>	Required. Must be one of the defined values: <code>PASSCODE_ONCE</code> , <code>PASSCODE_AIRWORD</code> , <code>PASSCODE_ALWAYS_REQUIRE</code> . <code>PASSCODE_ONCE</code> will require an on-screen passcode to be entered on the first connection from a device. Subsequent connections from the same device will not be prompted. <code>PASSCODE_ALWAYS_REQUIRE</code> will require an on-screen passcode to be entered upon every AirPlay connection. <code>PASSWORD</code> will require a passphrase to be entered as specified in the Password key. The Password key is required if this <code>SecurityType</code> is selected. <code>NO</code> is deprecated in tvOS 11.3. Existing profiles using <code>NO</code> will get the <code>PASSWORD_ALWAYS_REQUIRE</code> .	
<code>AccessType</code>	Required. Must be one of the defined values: <code>ALLOW_WIFIEONLY</code> , <code>ALLOW_WIFIANDAWDL</code> , <code>ALLOW_Ethernet</code> . <code>ALLOW_WIFIEONLY</code> connections only from devices on the same Ethernet/WiFi network as the Apple TV.	
<code>Password</code>	Optional. The AirPlay password. Required if <code>SecurityType</code> is <code>PASSWORD</code> .	

AirPlay Security Payload

The AirPlay Security payload locks the Apple TV to a particular style of AirPlay Security. The AirPlay Security payload is designated by specifying `com.apple.airplaySecurityPayload`.

This payload is supported on tvOS 11.0 and later.

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

Key	Type	Value
<code>SecurityType</code>	Required. Must be one of the defined values: <code>PASSCODE_ONCE</code> , <code>PASSCODE_AIRWORD</code> , <code>PASSCODE_ALWAYS_REQUIRE</code> . <code>PASSCODE_ONCE</code> will require an on-screen passcode to be entered on the first connection from a device. Subsequent connections from the same device will not be prompted. <code>PASSCODE_ALWAYS_REQUIRE</code> will require an on-screen passcode to be entered upon every AirPlay connection. <code>PASSWORD</code> will require a passphrase to be entered as specified in the Password key. The Password key is required if this <code>SecurityType</code> is selected. <code>NO</code> is deprecated in tvOS 11.3. Existing profiles using <code>NO</code> will get the <code>PASSWORD_ALWAYS_REQUIRE</code> .	
<code>AccessType</code>	Required. Must be one of the defined values: <code>ALLOW_WIFIEONLY</code> , <code>ALLOW_WIFIANDAWDL</code> , <code>ALLOW_Ethernet</code> . <code>ALLOW_WIFIEONLY</code> connections only from devices on the same Ethernet/WiFi network as the Apple TV.	
<code>Password</code>	Optional. The AirPlay password. Required if <code>SecurityType</code> is <code>PASSWORD</code> .	

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

10

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

10

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.airstripedPrinterLoadValue`.

This payload is supported on iOS 7.0 and later and on macOS 10.10 and later.

Key	Type	Value
<code>AirPrint</code>	Array of Dictionaries	An array of AirPrint printers that should always be shown.

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

Key	Type	Value
<code>IP Address String</code>	The IP Address of the AirPrint destination.	
<code>Resource Path</code>	The Resource Path associated with the printer. This corresponds to the <code>rp</code> parameter of the <code>_ippss:Bonjour</code> record. For example:	
		<ul style="list-style-type: none"> <code>printers/Canon_MG5300_series</code> <code>printers/Xerox_Phaser_7600</code> <code>ipp/print</code> <code>Epson_IPP_Printer</code>
<code>Port</code>	Integer	Listening port of the AirPrint destination. If this key is not specified AirPrint will use the default port. Availability: Available only in iOS 11.0 and later.
<code>Force TLS</code>	Boolean	If true AirPrint connections are secured by Transport Layer Security (TLS). Default is false. Availability: Available only in iOS 11.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

11

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.airstripedPrinterLoadValue`.

This payload is supported on iOS 7.0 and later and on macOS 10.10 and later.

Key	Type	Value
<code>AirPrint</code>	Array of Dictionaries	An array of AirPrint printers that should always be shown.

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

Key	Type	Value
<code>IP Address String</code>	The IP Address of the AirPrint destination.	
<code>Resource Path</code>	The Resource Path associated with the printer. This corresponds to the <code>rp</code> parameter of the <code>_ippss:Bonjour</code> record. For example:	
		<ul style="list-style-type: none"> <code>printers/Canon_MG5300_series</code> <code>printers/Xerox_Phaser_7600</code> <code>ipp/print</code> <code>Epson_IPP_Printer</code>
<code>Port</code>	Integer	Listening port of the AirPrint destination. If this key is not specified AirPrint will use the default port. Availability: Available only in iOS 11.0 and later.
<code>Force TLS</code>	Boolean	If true AirPrint connections are secured by Transport Layer Security (TLS). Default is false. Availability: Available only in iOS 11.0 and later.

App Lock Payload

The App Lock payload is designated by specifying `com.apple.appLockLoadValue`. 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.

Note

You can't update any app while the device is locked in Single App Mode. You need to exit Single App Mode long enough to update apps as needed. During that time you should restrict the visible apps as much as possible, except for Settings and Phone and any other apps that cannot be blacklisted.

This payload is supported only in iOS 6.0 and later.

The payload contains the following key:

Key	Type	Value
<code>App</code>	Dictionary	A dictionary containing information about the app.

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

Key	Type	Value
<code>Identifier</code>	String	The bundle identifier of the application.
<code>Options</code>	Dictionary	Optional. Described below. Availability: Available only in iOS 7.0 and later.
<code>UserEnabledOpDictionary</code>	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):

App Lock Payload

The App Lock payload is designated by specifying `com.apple.appLockLoadValue`. 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.

Note

You can't update any app while the device is locked in Single App Mode. You need to exit Single App Mode long enough to update apps as needed. During that time you should restrict the visible apps as much as possible, except for Settings and Phone and any other apps that cannot be blacklisted.

This payload is supported only in iOS 6.0 and later.

The payload contains the following key:

Key	Type	Value
<code>App</code>	Dictionary	A dictionary containing information about the app.

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

Key	Type	Value
<code>Identifier</code>	String	The bundle identifier of the application.
<code>Options</code>	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):

Key	Type	Value
DisableTouch	Boolean	Optional. If true, the touch screen is disabled. Default is false. Also, available in tvOS 10.2 and later to lock the touch pad on the remote.
DisableDeviceRotation	Boolean	Optional. If true, device rotation sensing is disabled. Default is false.
DisableVolumeButtons	Boolean	Optional. If true, the volume buttons are disabled. Default is false.
DisableRingerSwitch	Boolean	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	Boolean	Optional. If true, the sleep/wake button is disabled. Default is false.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

12

Key	Type	Value
DisableTouch	Boolean	Optional. If true, the touch screen is disabled. Default is false. Also, available in tvOS 10.2 and later to lock the touch pad on the remote.
DisableDeviceRotation	Boolean	Optional. If true, device rotation sensing is disabled. Default is false.
DisableVolumeButtons	Boolean	Optional. If true, the volume buttons are disabled. Default is false.
DisableRingerSwitch	Boolean	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	Boolean	Optional. If true, the sleep/wake button is disabled. Default is false.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

12

Key	Type	Value
DisableAutoLock	Boolean	Optional. If true, the device will not automatically go to sleep after an idle period. Also, available in tvOS 10.2 and later.
EnableVoiceOver	Boolean	Optional. If true, VoiceOver is turned on. Default is false. Also, available in tvOS 10.2 and later.
EnableZoom	Boolean	Optional. If true, Zoom is turned on. Default is false. Also, available in tvOS 10.2 and later.
EnableInvertColors	Boolean	Optional. If true, Invert Colors is turned on. Default is false. Also, available in tvOS 10.2 and later.
EnableAssistiveTouch	Boolean	Optional. If true, AssistiveTouch is turned on. Default is false.
EnableSpeakSelection	Boolean	Optional. If true, Speak Selection is turned on. Default is false.
EnableMonoAudio	Boolean	Optional. If true, Mono Audio is turned on. Default is false.

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

Key	Type	Value
VoiceOver	Boolean	Optional. If true, allow VoiceOver adjustment. Default is false. Also, available in tvOS 10.2 and later.
Zoom	Boolean	Optional. If true, allow Zoom adjustment. Default is false. Also, available in tvOS 10.2 and later.
InvertColors	Boolean	Optional. If true, allow Invert Colors adjustment. Default is false. Also, available in tvOS 10.2 and later.
AssistiveTouch	Boolean	Optional. If true, allow AssistiveTouch adjustment. Default is false.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

13

Key	Type	Value
DisableAutoLock	Boolean	Optional. If true, the device will not automatically go to sleep after an idle period. Also, available in tvOS 10.2 and later.
EnableVoiceOver	Boolean	Optional. If true, VoiceOver is turned on. Default is false. Also, available in tvOS 10.2 and later.
EnableZoom	Boolean	Optional. If true, Zoom is turned on. Default is false. Also, available in tvOS 10.2 and later.
EnableInvertColors	Boolean	Optional. If true, Invert Colors is turned on. Default is false. Also, available in tvOS 10.2 and later.
EnableAssistiveTouch	Boolean	Optional. If true, AssistiveTouch is turned on. Default is false.
EnableSpeakSelection	Boolean	Optional. If true, Speak Selection is turned on. Default is false.
EnableMonoAudio	Boolean	Optional. If true, Mono Audio is turned on. Default is false.

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

Key	Type	Value
VoiceOver	Boolean	Optional. If true, allow VoiceOver adjustment. Default is false. Also, available in tvOS 10.2 and later.
Zoom	Boolean	Optional. If true, allow Zoom adjustment. Default is false. Also, available in tvOS 10.2 and later.
InvertColors	Boolean	Optional. If true, allow Invert Colors adjustment. Default is false. Also, available in tvOS 10.2 and later.
AssistiveTouch	Boolean	Optional. If true, allow AssistiveTouch adjustment. Default is false.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

13

AppStore Payload

The AppStore payload is designated by specifying `com.apple.appstorepayload` and `Value`. It establishes macOS AppStore restrictions and is supported on the User channel.

The payload contains the following keys:

Key	Type	Value
restrict-store-require-admin-to	Boolean	Optional. Restrict app installations to admin users. Available on macOS 10.9 and later.
restrict-store-softwareupdate-only	Boolean	Optional. Restrict app installations to software updates only. Available on macOS 10.10 and later.
restrict-store-disable-app Adoption	Boolean	Optional. Disable App Adoption by users. Available on macOS 10.10 and later.
DisableSoftwareUpdateNotifications	Boolean	Optional. Disable software update notifications. Available on macOS 10.10 and later.

AppStore Payload

The AppStore payload is designated by specifying `com.apple.appstorepayload` and `Value`. It establishes macOS AppStore restrictions and is supported on the User channel.

The payload contains the following keys:

Key	Type	Value
restrict-store-require-admin-to	Boolean	Optional. Restrict app installations to admin users. Available on macOS 10.9 and later.
restrict-store-softwareupdate-only	Boolean	Optional. Restrict app installations to software updates only. Available on macOS 10.10 and later.
restrict-store-disable-app Adoption	Boolean	Optional. Disable App Adoption by users. Available on macOS 10.10 and later.
DisableSoftwareUpdateNotifications	Boolean	Optional. Disable software update notifications. Available on macOS 10.10 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

14

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

14

Autonomous Single App Mode

The payload is designated by specifying `com.apple.assessmentpayloadtype`

This payload grants Autonomous Single App Mode capabilities for specific applications. Available in macOS 10.13.4 and later.

It must be installed as a device profile. Only one payload of this type can be installed on a system. This payload can only be installed via a “user approved” MDM server.

Note

Applications listed in this payload will have low-level access to the system, including, but not limited to, key logging and user interface manipulation outside of the application’s context.

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

Key	Type	Value
<code>AllowedApplications</code>	Array of dictionaries that specify applications that are to be granted access to Assessment APIs.	

Each dictionary in the `AllowedApplications` consists of:

Key	Type	Value
<code>BundleIdentifier</code>	String	The application’s bundle identifier. <code>BundleIdentifier</code> must be unique. If two dictionaries contain the same <code>BundleIdentifier</code> but different <code>TeamIdentifier</code> , this will be considered a hard error and the payload will not be installed.
<code>TeamIdentifier</code>	String	The developer’s team identifier used to sign the application.

To be granted access, applications must be signed with the specified bundle identifier and team identifier using an Apple-issued production developer certificate. Applications must specify the `com.apple.developer.assessment` entitlement with a value of `true`.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

15

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

15

CalDAV Payload

The payload is designated by specifying `com.apple.caldavspayloadtype`

This payload configures a CalDAV account.

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

Key	Type	Value
<code>CalDAVAccountDescription</code>	String	Optional. The description of the account.
<code>CalDAVHostName</code>	String	The server address. In macOS, this key is required.
<code>CalDAVUsername</code>	String	The user’s login name. In macOS, this key is required.
<code>CalDAVPassword</code>	String	Optional. The user’s password.
<code>CalDAVUseSSL</code>	Boolean	Whether or not to use SSL. In macOS, this key is optional.
<code>CalDAVPort</code>	Integer	Optional. The port on which to connect to the server.
<code>CalDAVPrincipalURL</code>	String	Optional. The base URL to the user’s calendar. In macOS this URL is required if the user doesn’t provide a password, because auto-discovery of the service will fail and the account won’t be created.

Calendar Subscription Payload

The calendar subscription payload is designated by specifying `com.apple.subscribedcalendaraccount`

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

Autonomous Single App Mode

The payload is designated by specifying `com.apple.assessmentpayloadtype`

This payload grants Autonomous Single App Mode capabilities for specific applications. Available in macOS 10.13.4 and later.

It must be installed as a device profile. Only one payload of this type can be installed on a system. This payload can only be installed via a “user approved” MDM server.

Note

Applications listed in this payload will have low-level access to the system, including, but not limited to, key logging and user interface manipulation outside of the application’s context.

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

Key	Type	Value
<code>AllowedApplications</code>	Array of dictionaries that specify applications that are to be granted access to Assessment APIs.	

Each dictionary in the `AllowedApplications` consists of:

Key	Type	Value
<code>BundleIdentifier</code>	String	The application’s bundle identifier. <code>BundleIdentifier</code> must be unique. If two dictionaries contain the same <code>BundleIdentifier</code> but different <code>TeamIdentifier</code> , this will be considered a hard error and the payload will not be installed.
<code>TeamIdentifier</code>	String	The developer’s team identifier used to sign the application.

To be granted access, applications must be signed with the specified bundle identifier and team identifier using an Apple-issued production developer certificate. Applications must specify the `com.apple.developer.assessment` entitlement with a value of `true`.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

15

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

15

CalDAV Payload

The payload is designated by specifying `com.apple.caldavspayloadtype`

This payload configures a CalDAV account.

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

Key	Type	Value
<code>CalDAVAccountDescription</code>	String	Optional. The description of the account.
<code>CalDAVHostName</code>	String	The server address. In macOS, this key is required.
<code>CalDAVUsername</code>	String	The user’s login name. In macOS, this key is required.
<code>CalDAVPassword</code>	String	Optional. The user’s password.
<code>CalDAVUseSSL</code>	Boolean	Whether or not to use SSL. In macOS, this key is optional.
<code>CalDAVPort</code>	Integer	Optional. The port on which to connect to the server.
<code>CalDAVPrincipalURL</code>	String	Optional. The base URL to the user’s calendar. In macOS this URL is required if the user doesn’t provide a password, because auto-discovery of the service will fail and the account won’t be created.

Calendar Subscription Payload

The calendar subscription payload is designated by specifying `com.apple.subscribedcalendaraccount`

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

Username	String	Optional. A user name used for authentication.
Password	String	Optional. A password used for authentication.

Each A P Dictionary contains the following keys:

Key	Type	Value
Name	String	Required. The Access Point Name.
AuthenticationType	String	Optional. Must contain either C H A D P A P Defaults to P A P
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.
ProxyPort	Integer	Optional. The proxy server's port.
DefaultProtocolMas	Integer	Deprecated. Default Internet Protocol versions. Set to the same value as A l l o w e d P r o t o .PossibleValues are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.
AllowedProtocolMas	Integer	Optional. Supported Internet Protocol versions. Possible values are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

18

Username	String	Optional. A user name used for authentication.
Password	String	Optional. A password used for authentication.

Each A P Dictionary contains the following keys:

Key	Type	Value
Name	String	Required. The Access Point Name.
AuthenticationType	String	Optional. Must contain either C H A D P A P Defaults to P A P
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.
ProxyPort	Integer	Optional. The proxy server's port.
DefaultProtocolMas	Integer	Deprecated. Default Internet Protocol versions. Set to the same value as A l l o w e d P r o t o .PossibleValues are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.
AllowedProtocolMas	Integer	Optional. Supported Internet Protocol versions. Possible values are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

18

Key	Type	Value
AllowedProtocolMas	Integer	Optional. Supported Internet Protocol versions while roaming. Possible values are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.
InRoaming		
AllowedProtocolMas	Integer	Optional. Supported Internet Protocol versions while domestic roaming. Possible values are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.
InDomesticRoaming		

Certificate Payload

The P a y l o a d o f a certificate payload must be one of the following:

Payload type	Container format	Certificate type
com.apple.security.PKCS#8(.cert)	Alias for com.apple.security.pkcs1	
com.apple.security.PKCS#8(.cert)1	DER-encoded certificate without private key. May contain root certificates.	
com.apple.security.PKCS#8(.user)	PEM-encoded certificate without private key. May contain root certificates.	
com.apple.security.PKCS#12(.p12)	2 Password-protected identity certificate. Only one certificate may be included.	

In addition to the settings common to all payloads, all Certificate payloads define the following keys:

Key	Type	Value
PayloadCertificate	String	Optional. The file name of the enclosed certificate.
PayloadContent	Data	Mandatory. The base64 representation of the payload with a line length of 52.
Password	String	Optional. For PKCS#12 certificates, contains the password to the identity.
AllowAllAppsAccess	Boolean	Optional. For PKCS#12 certificates, if true, all apps have access to the private key. Default is false Availability: Available in macOS 10.10 and later.

Note

Because the password string is stored in the clear in the profile, it is recommended that the profile be encrypted for the device.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

19

Key	Type	Value
AllowedProtocolMas	Integer	Optional. Supported Internet Protocol versions while roaming. Possible values are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.
InRoaming		
AllowedProtocolMas	Integer	Optional. Supported Internet Protocol versions while domestic roaming. Possible values are: 1 = IPv4, 2 = IPv6, and 3 = Both. Availability: Available in iOS 10.3 and later.
InDomesticRoaming		

Certificate Payload

The P a y l o a d o f a certificate payload must be one of the following:

Payload type	Container format	Certificate type
com.apple.security.PKCS#8(.cert)	Alias for com.apple.security.pkcs1	
com.apple.security.PKCS#8(.cert)1	DER-encoded certificate without private key. May contain root certificates.	
com.apple.security.PKCS#8(.user)	PEM-encoded certificate without private key. May contain root certificates.	
com.apple.security.PKCS#12(.p12)	2 Password-protected identity certificate. Only one certificate may be included.	

In addition to the settings common to all payloads, all Certificate payloads define the following keys:

Key	Type	Value
PayloadCertificate	String	Optional. The file name of the enclosed certificate.
PayloadContent	Data	Mandatory. The base64 representation of the payload with a line length of 52.
Password	String	Optional. For PKCS#12 certificates, contains the password to the identity.
AllowAllAppsAccess	Boolean	Optional. For PKCS#12 certificates, if true, all apps have access to the private key. Default is false Availability: Available in macOS 10.10 and later.

Note

Because the password string is stored in the clear in the profile, it is recommended that the profile be encrypted for the device.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

19

Certificate Preference Payload

Certificate Preference payloads are designated by specifying com.apple.security.certificatepre as the P a y l o a d o f a Value. See also [Identity Preference Payload](#) for setting up identity preferences.

A Certificate Preference payload lets you identify a Certificate Preference item in the user's keychain that references a certificate payload included in the same profile. It can only appear in a user profile, not a device profile. You can include multiple Certificate Preference payloads as needed.

Available in Mac OS 10.12 and later.

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

Key	Type	Value
Name	String	Required. An email address (RFC822) or other name for which a preferred certificate is requested.
PayloadCertificateString	String	U U I D The UUID of another payload within the same profile that installed the certificate; for example, a 'com.apple.security.root' payload.

Certificate Preference Payload

Certificate Preference payloads are designated by specifying com.apple.security.certificatepre as the P a y l o a d o f a Value. See also [Identity Preference Payload](#) for setting up identity preferences.

A Certificate Preference payload lets you identify a Certificate Preference item in the user's keychain that references a certificate payload included in the same profile. It can only appear in a user profile, not a device profile. You can include multiple Certificate Preference payloads as needed.

Available in Mac OS 10.12 and later.

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

Key	Type	Value
Name	String	Required. An email address (RFC822) or other name for which a preferred certificate is requested.
PayloadCertificateString	String	U U I D The UUID of another payload within the same profile that installed the certificate; for example, a 'com.apple.security.root' payload.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

20

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

20

Certificate Transparency Payload

CertificatePreference payloads are designated by specifying `com.apple.security.certificateTransparencyPayload` as the `payloadValue`.

A Certificate Transparency payload controls [Certificate Transparency](#) enforcement. It can only appear in a device profile, not a user profile. You can include multiple Certificate Transparency payloads as needed.

This payload does not require MDM or DEP enrollment.

Available in iOS 12.1.1, MacOS 10.14.2, tvOS 12.1.1, and watchOS 5.1.1 and later.

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

Key	Type	Value
<code>DisabledForDomains</code>	String	Optional. A list of domains where certificate transparency is disabled. A leading period can be used to match subdomains, but a domain matching rule must not match all domains within a top level domain ("example.com" and "example.co.uk" are allowed while ".com" and ".co.uk" are not allowed).
<code>DisabledForCertificates</code>	Dictionary	Optional. A list of hashed <code>subjectPublicKeyInfo</code> defining the certificates where certificate transparency is disabled. For certificate transparency enforcement to be disabled, one of the following conditions must be met: <ul style="list-style-type: none"> The hash is of the server certificate's <code>subjectPublicKeyInfo</code> The hash is of a <code>subjectPublicKeyInfo</code> CA certificate in the certificate chain, that CA certificate is constrained via the X.509v3 nameConstraints extension, one or more <code>directoryName nameConstraints</code> are present in the permittedSubtrees, and the <code>directoryName</code> contains an <code>organizationName</code> attribute. The hash is of a <code>subjectPublicKeyInfo</code> CA certificate in the certificate chain, the CA certificate has one or more <code>organizationName</code> attributes in the certificate Subject, and the server's certificate contains the same number of <code>organizationName</code> attributes, in the same order, and with byte-for-byte identical values.

The `hashedSubjectPublicKeyInfo` contains the following keys:

Key	Type	Value
<code>Algorithm</code>	String	Required. Currently, must be "sha256".
<code>Hash</code>	Data	Required. Created by applying the specified hash algorithm to the DER-encoding of the certificate's <code>subjectPublicKeyInfo</code> .

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

21

To generate the data specified by the `Hash` in the `subjectPublicKeyInfo` use this command for a PEM encoded certificate:

```
openssl x509 -pubkey -in example_certificate.pem -inform pem | openssl dgst -sha256 -binary | base64
```

If your certificate is DER encoded, use this command:

```
openssl x509 -pubkey -in example_certificate.der -inform der | openssl dgst -sha256 -binary | base64
```

If your certificate does not have a `.pem` or `.der` extension, use the `file` command to identify its encoding type.

```
$ file example_certificate.crt
example_certificate.crt: PEM certificate
$ file example_certificate.cer
example_certificate.cer: data
```

Conference Room Display Payload

Certificate Transparency Payload

CertificatePreference payloads are designated by specifying `com.apple.security.certificateTransparency` as the `payloadValue`.

A Certificate Transparency payload controls [Certificate Transparency](#) enforcement. It can only appear in a device profile, not a user profile. You can include multiple Certificate Transparency payloads as needed.

This payload does not require MDM or DEP enrollment.

Available in iOS 12.1.1, MacOS 10.14.2, tvOS 12.1.1, and watchOS 5.1.1 and later.

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

Key	Type	Value
<code>DisabledForDomains</code>	String	Optional. A list of domains where certificate transparency is disabled. A leading period can be used to match subdomains, but a domain matching rule must not match all domains within a top level domain ("example.com" and "example.co.uk" are allowed while ".com" and ".co.uk" are not allowed).
<code>DisabledForCertificates</code>	Dictionary	Optional. A list of hashed <code>subjectPublicKeyInfo</code> defining the certificates where certificate transparency is disabled. For certificate transparency enforcement to be disabled, one of the following conditions must be met: <ul style="list-style-type: none"> The hash is of the server certificate's <code>subjectPublicKeyInfo</code> The hash is of a <code>subjectPublicKeyInfo</code> CA certificate in the certificate chain, that CA certificate is constrained via the X.509v3 nameConstraints extension, one or more <code>directoryName nameConstraints</code> are present in the permittedSubtrees, and the <code>directoryName</code> contains an <code>organizationName</code> attribute. The hash is of a <code>subjectPublicKeyInfo</code> CA certificate in the certificate chain, the CA certificate has one or more <code>organizationName</code> attributes in the certificate Subject, and the server's certificate contains the same number of <code>organizationName</code> attributes, in the same order, and with byte-for-byte identical values.

The `hashedSubjectPublicKeyInfo` contains the following keys:

Key	Type	Value
<code>Algorithm</code>	String	Required. Currently, must be "sha256".
<code>Hash</code>	Data	Required. Created by applying the specified hash algorithm to the DER-encoding of the certificate's <code>subjectPublicKeyInfo</code> .

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

21

To generate the data specified by the `Hash` in the `subjectPublicKeyInfo` use this command for a PEM encoded certificate:

```
openssl x509 -pubkey -in example_certificate.pem -inform pem | openssl dgst -sha256 -binary | base64
```

If your certificate is DER encoded, use this command:

```
openssl x509 -pubkey -in example_certificate.der -inform der | openssl dgst -sha256 -binary | base64
```

If your certificate does not have a `.pem` or `.der` extension, use the `file` command to identify its encoding type.

```
$ file example_certificate.crt
example_certificate.crt: PEM certificate
$ file example_certificate.cer
example_certificate.cer: data
```

Conference Room Display Payload

The Conference Room Display payload is designated by specifying `com.apple.conferenceRoomDisplayPayload`

`Type`

It configures an Apple TV to enter Conference Room Display mode and restricts exit from that mode. It is supported on supervised devices running tvOS 10.2 or later.

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

Key	Type	Value
<code>MessageString</code>	Optional. A custom message displayed on the screen in Conference Room Display mode.	

Note

When Conference Room Display mode and Single App mode are both enabled, Conference Room Display mode is active and the user can't access the app.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

22

The Conference Room Display payload is designated by specifying `com.apple.conferenceRoomDisplayPayload`

`Type`

It configures an Apple TV to enter Conference Room Display mode and restricts exit from that mode. It is supported on supervised devices running tvOS 10.2 or later.

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

Key	Type	Value
<code>MessageString</code>	Optional. A custom message displayed on the screen in Conference Room Display mode.	

Note

When Conference Room Display mode and Single App mode are both enabled, Conference Room Display mode is active and the user can't access the app.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

22

Content Caching Payload

The Content Caching payload is designated by specifying `com.apple.AssetCachePayload`

It configures the Content Caching service.

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

Key	Type	Value
<code>AllowPersonalCache</code>	Boolean	Optional. If set to <code>true</code> , caches the user's iCloud data. Clients may take some time (hours, days) to react to changes to this setting; it does not have an immediate effect. Default is <code>true</code> . At least one of the <code>AllowPersonalCache</code> and <code>AllowSharedCache</code> keys must be <code>true</code> . Availability: Available in macOS 10.13.4 and later.
<code>AllowSharedCache</code>	Boolean	Optional. If set to <code>true</code> , caches non-iCloud content, such as apps and software updates. Clients may take some time (hours, days) to react to changes to this setting; it does not have an immediate effect. Default is <code>true</code> . At least one of the <code>AllowPersonalCache</code> and <code>AllowSharedCache</code> keys must be <code>true</code> . Availability: Available in macOS 10.13.4 and later.
<code>AutoActivation</code>	Boolean	Optional. If set to <code>true</code> , automatically activate the Content Cache when possible and prevent disabling of the Content Cache. Default is <code>false</code> . Availability: Available in macOS 10.13.4 and later.
<code>CacheLimit</code>	Integer	Optional. Defines the maximum number of bytes of disk space that will be used for the Content Cache. A <code>CacheLimit</code> of 0 means unlimited disk space. Default is 0. Availability: Available in macOS 10.13.4 and later.
<code>FilePath</code>	String	Optional. The path to the directory used to store Cached Content. Changing this setting manually does not automatically move cached content from the old to the new location. To move content automatically, use the Sharing preference's Content Caching pane. The value must be, or end with, <code>/Library/Application Support/Apple/AssetCache</code> (and list all intermediates) will be created for the given <code>FilePath</code> if it does not already exist. The directory will be owned by <code>_assetcache:_asadminmode0750</code> . Its immediate parent directory <code>.. / Library/Application Support/Apple/AssetCache</code> will be owned by <code>_assetcache:_asadminmode0755</code> . Default is <code>/Library/Application Support/Apple/AssetCache/Cache</code> . Availability: Available in macOS 10.13.4 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

23

Content Caching Payload

The Content Caching payload is designated by specifying `com.apple.AssetCachePayload`

It configures the Content Caching service.

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

Key	Type	Value
<code>AllowPersonalCache</code>	Boolean	Optional. If set to <code>true</code> , caches the user's iCloud data. Clients may take some time (hours, days) to react to changes to this setting; it does not have an immediate effect. Default is <code>true</code> . At least one of the <code>AllowPersonalCache</code> and <code>AllowSharedCache</code> keys must be <code>true</code> . Availability: Available in macOS 10.13.4 and later.
<code>AllowSharedCache</code>	Boolean	Optional. If set to <code>true</code> , caches non-iCloud content, such as apps and software updates. Clients may take some time (hours, days) to react to changes to this setting; it does not have an immediate effect. Default is <code>true</code> . At least one of the <code>AllowPersonalCache</code> and <code>AllowSharedCache</code> keys must be <code>true</code> . Availability: Available in macOS 10.13.4 and later.
<code>AutoActivation</code>	Boolean	Optional. If set to <code>true</code> , automatically activate the Content Cache when possible and prevent disabling of the Content Cache. Default is <code>false</code> . Availability: Available in macOS 10.13.4 and later.
<code>CacheLimit</code>	Integer	Optional. Defines the maximum number of bytes of disk space that will be used for the Content Cache. A <code>CacheLimit</code> of 0 means unlimited disk space. Default is 0. Availability: Available in macOS 10.13.4 and later.
<code>FilePath</code>	String	Optional. The path to the directory used to store Cached Content. Changing this setting manually does not automatically move cached content from the old to the new location. To move content automatically, use the Sharing preference's Content Caching pane. The value must be, or end with, <code>/Library/Application Support/Apple/AssetCache</code> (and list all intermediates) will be created for the given <code>FilePath</code> if it does not already exist. The directory will be owned by <code>_assetcache:_asadminmode0750</code> . Its immediate parent directory <code>.. / Library/Application Support/Apple/AssetCache</code> will be owned by <code>_assetcache:_asadminmode0755</code> . Default is <code>/Library/Application Support/Apple/AssetCache/Cache</code> . Availability: Available in macOS 10.13.4 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

23

Key	Type	Value
<code>DenyTetheredCache</code>	Boolean	Optional. If set to <code>true</code> , tethered caching is disabled. Default is <code>false</code> . Availability: Available in macOS 10.13.4 and later.
<code>ListenRanges</code>	Array of Dictionaries	Optional. Array of dictionaries describing a range of client IP addresses to serve. Availability: Available in macOS 10.13.4 and later.
<code>ListenRangesOnly</code>	Boolean	Optional. If set to <code>true</code> , the Content Cache provides content only to clients in the ranges specified by the <code>ListenRanges</code> key. To use the <code>ListenRanges</code> key, the <code>ListenRanges</code> key must also be specified. Default is <code>false</code> . Availability: Available in macOS 10.13.4 and later.
<code>ListenWithPeers</code>	Boolean	Optional. If set to <code>true</code> , the Content Cache provides content to the clients in the union of the <code>ListenRanges</code> , <code>PeerListenRanges</code> and <code>Parents</code> ranges. Default is <code>true</code> . Availability: Available in macOS 10.13.4 and later.

Key	Type	Value
<code>DenyTetheredCache</code>	Boolean	Optional. If set to <code>true</code> , tethered caching is disabled. Default is <code>false</code> . Availability: Available in macOS 10.13.4 and later.
<code>ListenRanges</code>	Array of Dictionaries	Optional. Array of dictionaries describing a range of client IP addresses to serve. Availability: Available in macOS 10.13.4 and later.
<code>ListenRangesOnly</code>	Boolean	Optional. If set to <code>true</code> , the Content Cache provides content only to clients in the ranges specified by the <code>ListenRanges</code> key. To use the <code>ListenRanges</code> key, the <code>ListenRanges</code> key must also be specified. Default is <code>false</code> . Availability: Available in macOS 10.13.4 and later.
<code>ListenWithPeers</code>	Boolean	Optional. If set to <code>true</code> , the Content Cache provides content to the clients in the union of the <code>ListenRanges</code> , <code>PeerListenRanges</code> and <code>Parents</code> ranges. Default is <code>true</code> . Availability: Available in macOS 10.13.4 and later.

LocalSubnetsOnly Boolean	Optional. If set to true, the Content Cache offers content to clients only on the same immediate local network as the Content Cache. No content would be offered to clients on other networks reachable by the Content Cache. Default is true if LocalSubnets is set to true, ListenRaw will be ignored.
LogClientIdentit Boolean	Optional. If set to true, the Content Cache will log the IP address and port number of the clients that request content. Default is false.
Parents	Optional. Array of the local IP addresses of other Content Caches that this cache should download from or upload to, instead of downloading from or uploading to Apple directly. Invalid addresses and addresses of computers that are not Content Caches are ignored. Parent caches that become unavailable are skipped. If all parent Content Caches become unavailable, the Content Cache will download from or upload to Apple directly until a parent Content Cache becomes available again.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

24

LocalSubnetsOnly Boolean	Optional. If set to true, the Content Cache offers content to clients only on the same immediate local network as the Content Cache. No content would be offered to clients on other networks reachable by the Content Cache. Default is true if LocalSubnets is set to true, ListenRaw will be ignored.
LogClientIdentit Boolean	Optional. If set to true, the Content Cache will log the IP address and port number of the clients that request content. Default is false.
Parents	Optional. Array of the local IP addresses of other Content Caches that this cache should download from or upload to, instead of downloading from or uploading to Apple directly. Invalid addresses and addresses of computers that are not Content Caches are ignored. Parent caches that become unavailable are skipped. If all parent Content Caches become unavailable, the Content Cache will download from or upload to Apple directly until a parent Content Cache becomes available again.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

24

Key	Type	Value
ParentSelectionPriority	Optional. The policy to use when choosing among more than one configured parent Content Cache. With every policy, parent caches that are temporarily unavailable are skipped. <ul style="list-style-type: none"> first-available: the first parent in the Parents list that is available. This is useful for designating permanent primary, secondary, and subsequent parents. url-path-Hash: path part of the requested URL so that the same parent is always used for the same URL. This is useful for maximizing the size of the combined caches of the parents. random: choose a parent at random. This is useful for load balancing. round-robin: through the parents in order. This is useful for load balancing. sticky-available: with the first parent in the Parents list, always use the first parent that is available. Use that parent until it becomes unavailable, then advance to the next one. This is useful for designating floating primary, secondary, and subsequent parents. Default is round-robin.	
Availability:	Available in macOS 10.13.4 and later.	
PeerFilterRanges	Optional. Array of dictionaries describing a range of peer IP addresses that the Content Cache will use to filter its list of peers to query for content. The Content Cache only queries peers that are in the PeerFilterRanges. An empty array the Content Cache will not query any peers.	
Availability:	Available in macOS 10.13.4 and later.	
PeerListenRanges	Optional. Array of dictionaries describing a range of peer IP addresses the Content Cache will respond to peer cache queries from. When PeerListenRanges is an empty array, the Content Cache will respond with an error to all cache queries.	
Availability:	Available in macOS 10.13.4 and later.	
PeerLocalSubnets	Optional. If set to true, the Content Cache will only peer with other Content Caches on the same immediate local network, rather than with Content Caches that use the same public IP address as the device. When PeerLocalSubnets is set to true, it overrides the configuration of PeerFilterRanges. PeerListenRanges changes, the local network peering restrictions update appropriately.	
Availability:	Available in macOS 10.13.4 and later.	

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

25

Key	Type	Value
ParentSelectionPriority	Optional. The policy to use when choosing among more than one configured parent Content Cache. With every policy, parent caches that are temporarily unavailable are skipped. <ul style="list-style-type: none"> first-available: the first parent in the Parents list that is available. This is useful for designating permanent primary, secondary, and subsequent parents. url-path-Hash: path part of the requested URL so that the same parent is always used for the same URL. This is useful for maximizing the size of the combined caches of the parents. random: choose a parent at random. This is useful for load balancing. round-robin: through the parents in order. This is useful for load balancing. sticky-available: with the first parent in the Parents list, always use the first parent that is available. Use that parent until it becomes unavailable, then advance to the next one. This is useful for designating floating primary, secondary, and subsequent parents. Default is round-robin.	
Availability:	Available in macOS 10.13.4 and later.	
PeerFilterRanges	Optional. Array of dictionaries describing a range of peer IP addresses that the Content Cache will use to filter its list of peers to query for content. The Content Cache only queries peers that are in the PeerFilterRanges. An empty array the Content Cache will not query any peers.	
Availability:	Available in macOS 10.13.4 and later.	
PeerListenRanges	Optional. Array of dictionaries describing a range of peer IP addresses the Content Cache will respond to peer cache queries from. When PeerListenRanges is an empty array, the Content Cache will respond with an error to all cache queries.	
Availability:	Available in macOS 10.13.4 and later.	
PeerLocalSubnets	Optional. If set to true, the Content Cache will only peer with other Content Caches on the same immediate local network, rather than with Content Caches that use the same public IP address as the device. When PeerLocalSubnets is set to true, it overrides the configuration of PeerFilterRanges. PeerListenRanges changes, the local network peering restrictions update appropriately.	
Availability:	Available in macOS 10.13.4 and later.	

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

25

Key	Type	Value
Port	Integer	Optional. The TCP port number on which the Content Cache accepts requests for uploads or downloads. Port set to 0 picks a random, available port. Default is 0.
Availability:	Available in macOS 10.13.4 and later.	
PublicRanges	Optional. Array of dictionaries describing a range of public IP addresses that the cloud servers should use for matching clients to Content Caches.	
Availability:	Available in macOS 10.13.4 and later.	

The dictionary used to define ranges used by the Content Cache uses the following keys:

Key	Type	Value
-----	------	-------

Key	Type	Value
Port	Integer	Optional. The TCP port number on which the Content Cache accepts requests for uploads or downloads. Port set to 0 picks a random, available port. Default is 0.
Availability:	Available in macOS 10.13.4 and later.	
PublicRanges	Optional. Array of dictionaries describing a range of public IP addresses that the cloud servers should use for matching clients to Content Caches.	
Availability:	Available in macOS 10.13.4 and later.	

The dictionary used to define ranges used by the Content Cache uses the following keys:

Key	Type	Value
-----	------	-------

Key	Type	Value
type	String	Optional. The IP address type (IPv4 or IPv6). Default is IPv4.
first	String	Required. First IP address in the range.
last	String	Required. Last IP address in the range.

Key	Type	Value
type	String	Optional. The IP address type (IPv4 or IPv6). Default is IPv4.
first	String	Required. First IP address in the range.
last	String	Required. Last IP address in the range.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

26

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

26

Desktop Payload

The Desktop payload is designated by specifying `com.apple.desktoppayload.type`

This payload sets up macOS Desktop settings and restrictions. It is supported on the user channel and on macOS 10.10 and later.

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

Key	Type	Value
locked	Boolean	Optional. If true, the desktop picture is locked. Default is false.
override-picture-path	String	Optional. If supplied, it sets the path to the desktop picture.

DNS Proxy Payload

The DNS Proxy payload is designated by specifying `com.apple.dnsProxyPayload.type`. This payload can be installed only on a Supervised device.

This payload sets up iOS DNS Proxy settings. It is supported on iOS 11 and later.

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

Key	Type	Value
AppBundleIdentifier	String	Required. Bundle identifier of the app containing the DNS proxy network extension.
ProviderBundleIdentifier	String	Optional. Bundle identifier of the DNS proxy network extension to use. Useful for apps that contain more than one DNS proxy extension.
ProviderConfigurationDictionary	Dictionary	Optional. Dictionary of vendor-specific configuration items.

Dock Payload

The Dock payload is designated by specifying `com.apple.desklyoad.type`

The Dock payload is supported on the user channel and, except for `AllowDockFixup`, on all versions of macOS. The key `AllowDockFixup` is supported on macOS 10.12 and later.

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

Key	Type	Value
orientation	String	Optional. Orientation of the dock. Values may be bottom, left, right.
position-immutable	Boolean	Optional. If true, the position is locked.
autohide	Boolean	Optional. If true, automatically hide and show the dock.
autohide-immutable	Boolean	Optional. If true, the Automatically Hide checkbox is disabled.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

27

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

27

Key	Type	Value
minimize-to-application	Boolean	Optional. If true, enable the minimize-to-application feature.
magnification	Boolean	Optional. If true, magnification is active.
magnify-immutable	Boolean	Optional. If true, the magnification checkbox is disabled.
large-size	Integer	Optional. The size of the largest magnification. Values must be in range 16 to 128.
mag-size-immutable	Boolean	Optional. If true, the magnify slider is disabled.
show-process-indicator	Boolean	Optional. If true, show the process indicator.

Key	Type	Value
minimize-to-application	Boolean	Optional. If true, enable the minimize-to-application feature.
magnification	Boolean	Optional. If true, magnification is active.
magnify-immutable	Boolean	Optional. If true, the magnification checkbox is disabled.
large-size	Integer	Optional. The size of the largest magnification. Values must be in range 16 to 128.
mag-size-immutable	Boolean	Optional. If true, the magnify slider is disabled.
show-process-indicator	Boolean	Optional. If true, show the process indicator.

launchnanim	Boolean	Optional. If true, animate opening applications.
launchnanim-immutable	Boolean	Optional. If true, the Animate Opening Applications checkbox is disabled.
mineffect	String	Optional. Set minimize effect. Values may be genie or scale
mineffect-immutable	Boolean	Optional. If true, the Minimize Using popup is disabled.
tilesize	Integer	Optional. The tile size. Values must be in range 16 to 128.
size-immutable	Boolean	Optional. If true, the size slider will be disabled.
MCXDockSpecialFolders	Array of Strings	Optional. One or more special folders that may be created at user login time and placed in the dock. Values may be AddDockMCXMyApplicationsFolder AddDockMCXDocumentsFolder AddDockMCXSharedFolder AddDockMCXOriginalNetworkHomeFolder "My Applications" item is only used for Simple Finder environments. The "Original Network Home" item is only used for mobile account users.
AllowDockFixupOverride	Boolean	Optional. If true, use the file in /Library/Preferences/com.apple.dockfixup.plist or migrated user logs in. The format of this file currently has no documentation. This option has no effect for existing users.
static-only	Boolean	Optional. If true, the device will use the static-apps and static-others dictionaries for the dock and ignore any items in the persistent-apps and persistent-others dictionaries. If a list the contents will be merged with the static items listed first.
static-others	Array of Dictionaries	Optional. Dock items in the Documents side that cannot be removed from the dock.
static-apps	Array of Dictionaries	Optional. Dock items in the Applications side that cannot be removed from the dock.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

28

launchnanim	Boolean	Optional. If true, animate opening applications.
launchnanim-immutable	Boolean	Optional. If true, the Animate Opening Applications checkbox is disabled.
mineffect	String	Optional. Set minimize effect. Values may be genie or scale
mineffect-immutable	Boolean	Optional. If true, the Minimize Using popup is disabled.
tilesize	Integer	Optional. The tile size. Values must be in range 16 to 128.
size-immutable	Boolean	Optional. If true, the size slider will be disabled.
MCXDockSpecialFolders	Array of Strings	Optional. One or more special folders that may be created at user login time and placed in the dock. Values may be AddDockMCXMyApplicationsFolder AddDockMCXDocumentsFolder AddDockMCXSharedFolder AddDockMCXOriginalNetworkHomeFolder "My Applications" item is only used for Simple Finder environments. The "Original Network Home" item is only used for mobile account users.
AllowDockFixupOverride	Boolean	Optional. If true, use the file in /Library/Preferences/com.apple.dockfixup.plist or migrated user logs in. The format of this file currently has no documentation. This option has no effect for existing users.
static-only	Boolean	Optional. If true, the device will use the static-apps and static-others dictionaries for the dock and ignore any items in the persistent-apps and persistent-others dictionaries. If a list the contents will be merged with the static items listed first.
static-others	Array of Dictionaries	Optional. Dock items in the Documents side that cannot be removed from the dock.
static-apps	Array of Dictionaries	Optional. Dock items in the Applications side that cannot be removed from the dock.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

28

Key	Type	Value
contents-immutable	Boolean	Optional. If true, the user cannot remove any item from or add any item to the dock.

The static-others static-dictionaries define the following keys:

Key	Type	Value
tile-data	Dictionary	Required. Information about a dock item.
tile-type	String	Required. The type of the tile. Values may be file-tile, directory, or tile url - if you are unsure whether the file item is a file or a directory, set this key to file-tile

The tile-d dictionary defines the following keys:

Key	Type	Value
label	String	Required. Label of a dock item.
url	String	Optional. For URL tiles, the URL string.
file-type	Integer	Required. The type of the tile expressed as a number. 3 = direct, 0 = URL = file

Education Configuration Payload

The Education Configuration Payload is designated by specifying com.apple.educationPayload and Type value. It can contain only one payload.

The Education Configuration Payload defines the users, groups, and departments within an educational organization.

It is supported on iOS 9.3 and later. On iOS, it must be supervised and sent over the device channel.

It is supported on macOS 10.14 and later. On macOS, it must be sent over the user channel.

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

Key	Type	Value
OrganizationUUID	String	Required. The organization's UUID identifier. This can be any valid UUID. All teacher and student devices that need to communicate with one another must have the same OrganizationIdentifier if they originated from different Device Enrollment Programs.
OrganizationName	String	Required. The organization's display name. This name will be shown in the iOS login screen.
PayloadCertificate	String	Required. The UUID of an identity certificate payload that will be used to perform client authentication with other devices.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

29

Key	Type	Value
contents-immutable	Boolean	Optional. If true, the user cannot remove any item from or add any item to the dock.

The static-others static-dictionaries define the following keys:

Key	Type	Value
tile-data	Dictionary	Required. Information about a dock item.
tile-type	String	Required. The type of the tile. Values may be file-tile, directory, or tile url - if you are unsure whether the file item is a file or a directory, set this key to file-tile

The tile-d dictionary defines the following keys:

Key	Type	Value
label	String	Required. Label of a dock item.
url	String	Optional. For URL tiles, the URL string.
file-type	Integer	Required. The type of the tile expressed as a number. 3 = direct, 0 = URL = file

Education Configuration Payload

The Education Configuration Payload is designated by specifying com.apple.educationPayload and Type value. It can contain only one payload.

The Education Configuration Payload defines the users, groups, and departments within an educational organization.

It is supported on iOS 9.3 and later. On iOS, it must be supervised and sent over the device channel.

It is supported on macOS 10.14 and later. On macOS, it must be sent over the user channel. **Student payloads are only supported on macOS 10.14.4 and later.**

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

Key	Type	Value
OrganizationUUID	String	Required. The organization's UUID identifier. This can be any valid UUID. All teacher and student devices that need to communicate with one another must have the same OrganizationIdentifier if they originated from different Device Enrollment Programs.
OrganizationName	String	Required. The organization's display name. This name will be shown in the iOS login screen.
PayloadCertificate	String	Required. The UUID of an identity certificate payload that will be used to perform client authentication with other devices.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

29

Key	Type	Value
LeaderPayloadCertificate	String	Optional. An array of UUIDs referring to certificate payloads that will be used to authorize leader peer certificate identities. This array must contain all certificates needed to validate the entire chain of trust. Leader certificates must have the common name prefix lead (case insensitive) and have a .crt type.

MemberPayloadCertificates	Optional. An array of UUIDs referring to certificate payloads that will be used to authorize group member peer certificate identities. This array must contain all certificates needed to validate the entire chain of trust. Member certificates must have the common name prefix <code>emblem</code> (case insensitive) and have a <code>certType</code> .
ResourcePayloadCertificateUUID	String Optional. The UUID of an identity certificate payload that will be used to perform client authentication when fetching additional resources, such as, student images. If not specified the MDM client identity will be used.
UserID	String Optional. A unique string that identifies the user of this device within the organization.
Departments	Array Optional. Shared: An array of dictionaries that define departments that are shown in the iOS login window.
Groups	Array Required. Shared: An array of dictionaries that define groups that the user can select in the login window. Leader: An array of dictionaries that define the groups that the user can control. Member: An array of dictionaries that define the groups of which the user is a member.
Users	Array Required. Shared: An array of dictionaries that define the users that are shown in the iOS login window. Leader: An array of dictionaries that define users that are members of the leader's groups. Member: An array of dictionaries that must contain the definition of the user specified in the UserID key. With one-to-one member devices, this key should include only the device user and the leader but not other class members.
DeviceGroups	Array Optional. Leader: An array of dictionaries that define the device groups to which the leader can assign devices. This key is not included in member payloads.
ScreenObservationPermissionModificationAll	Boolean Optional. If set to <code>true</code> , students enrolled in managed classes can modify their teacher's permissions for screen observation on this device. Defaults to <code>false</code> .

The `Departments` key must contain an array of dictionaries with the following key-value pairs:

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

30

MemberPayloadCertificates	Optional. An array of UUIDs referring to certificate payloads that will be used to authorize group member peer certificate identities. This array must contain all certificates needed to validate the entire chain of trust. Member certificates must have the common name prefix <code>emblem</code> (case insensitive) and have a <code>certType</code> .
ResourcePayloadCertificateUUID	String Optional. The UUID of an identity certificate payload that will be used to perform client authentication when fetching additional resources, such as, student images. If not specified the MDM client identity will be used.
UserID	String Optional. A unique string that identifies the user of this device within the organization.
Departments	Array Optional. Shared: An array of dictionaries that define departments that are shown in the iOS login window.
Groups	Array Required. Shared: An array of dictionaries that define groups that the user can select in the login window. Leader: An array of dictionaries that define the groups that the user can control. Member: An array of dictionaries that define the groups of which the user is a member.
Users	Array Required. Shared: An array of dictionaries that define the users that are shown in the iOS login window. Leader: An array of dictionaries that define users that are members of the leader's groups. Member: An array of dictionaries that must contain the definition of the user specified in the UserID key. With one-to-one member devices, this key should include only the device user and the leader but not other class members.
DeviceGroups	Array Optional. Leader: An array of dictionaries that define the device groups to which the leader can assign devices. This key is not included in member payloads.
ScreenObservationPermissionModificationAll	Boolean Optional. If set to <code>true</code> , students enrolled in managed classes can modify their teacher's permissions for screen observation on this device. Defaults to <code>false</code> .

The `Departments` key must contain an array of dictionaries with the following key-value pairs:

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

30

Key	Type	Value
Name	String	Required. The display name of the department.
GroupBeaconArbors	Required. The group beacon identifiers that are members of this department.	

The `Groups` key must contain an array of dictionaries with the following key-value pairs:

Key	Type	Value
BeaconID	Integer	Required. The unsigned 16 bit integer specifying this group's unique beacon ID.
Name	String	Required. The display name of the group.
Description	String	Optional. Description of the group.
ImageURL	String	Deprecated in iOS 9.3.1 and later. URL of an image for the group.
Configurations	String	Optional. The source that provided this group; e.g. iTunesU, SIS, or MDM.
LeaderIdentifiers	Array	Optional. The user identifiers that are leaders of this group.
MemberIdentifiers	Array	Required. The strings that refer to entries in the <code>Users</code> key that are members of the group.
DeviceGroupIdentifiers	Required. The identifier strings that refer to entries in the <code>DeviceGroups</code> key that are device groups to which the <code>instructor</code> can assign users from this class.	

The `Users` key must contain an array of dictionaries with the following key-value pairs:

Key	Type	Value
Identifier	String	Required. The string that uniquely identifies a user in the organization.
Name	String	Required. The string displayed as the name of the user.
GivenName	String	Optional. The string displayed as the given name of the user.
FamilyName	String	Optional. The string displayed as the family name of the user.
PhoneticGivenName	String	Optional. The string that represents the user's phonetic given name. It will be used to sort users in the Classroom app and the Shared iPad Login Screen.
PhoneticFamilyName	String	Optional. The string that represents the user's phonetic family name. It will be used to sort users in the Classroom app and the Shared iPad Login Screen.
ImageURL	String	Optional. A string containing a URL pointing to an image of the user. This image will be displayed in the iOS login screen and in the Classroom app. The recommended resolution is 256 x 256 pixels (512 x 512 pixels on a 2x device). The recommended formats are JPEG, PNG, and TIFF. The <code>ResourcePayloadCertificateID</code> or the MDM client identity will be used to perform authentication when fetching the image.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

31

Key	Type	Value
Name	String	Required. The display name of the department.
GroupBeaconArbors	Required. The group beacon identifiers that are members of this department.	

The `Groups` key must contain an array of dictionaries with the following key-value pairs:

Key	Type	Value
BeaconID	Integer	Required. The unsigned 16 bit integer specifying this group's unique beacon ID.
Name	String	Required. The display name of the group.
Description	String	Optional. Description of the group.
ImageURL	String	Deprecated in iOS 9.3.1 and later. URL of an image for the group.
Configurations	String	Optional. The source that provided this group; e.g. iTunesU, SIS, or MDM.
LeaderIdentifiers	Array	Optional. The user identifiers that are leaders of this group.
MemberIdentifiers	Array	Required. The strings that refer to entries in the <code>Users</code> key that are members of the group.
DeviceGroupIdentifiers	Required. The identifier strings that refer to entries in the <code>DeviceGroups</code> key that are device groups to which the teacher can assign users from this class.	

The `Users` key must contain an array of dictionaries with the following key-value pairs:

Key	Type	Value
Identifier	String	Required. The string that uniquely identifies a user in the organization.
Name	String	Required. The string displayed as the name of the user.
GivenName	String	Optional. The string displayed as the given name of the user.
FamilyName	String	Optional. The string displayed as the family name of the user.
PhoneticGivenName	String	Optional. The string that represents the user's phonetic given name. It will be used to sort users in the Classroom app and the Shared iPad Login Screen.
PhoneticFamilyName	String	Optional. The string that represents the user's phonetic family name. It will be used to sort users in the Classroom app and the Shared iPad Login Screen.
ImageURL	String	Optional. A string containing a URL pointing to an image of the user. This image will be displayed in the iOS login screen and in the Classroom app. The recommended resolution is 256 x 256 pixels (512 x 512 pixels on a 2x device). The recommended formats are JPEG, PNG, and TIFF. The <code>ResourcePayloadCertificateID</code> or the MDM client identity will be used to perform authentication when fetching the image.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

31

Key	Type	Value
FullScreenImageURL	Deprecated in iOS 9.3.1 and later. URL pointing to an image of the user. The <code>ResourcePayloadCertificateID</code> or the MDM client identity will be used to perform authentication when fetching the image.	

Key	Type	Value
FullScreenImageURL	Deprecated in iOS 9.3.1 and later. URL pointing to an image of the user. The <code>ResourcePayloadCertificateID</code> or the MDM client identity will be used to perform authentication when fetching the image.	

Apple ID	String	Specifying the specified resource.
Passcode Type	String	Optional. The Managed Apple ID for this user. Optional. The passcode UI to show when the user is at the login window; possible values are <code>com.apple.ersix</code>

The Device Group must contain an array of dictionaries with the following key-value pairs:

Key	Type	Content
Identifier	String	Required. The string that uniquely identifies the device group in the organization.
Name	String	Required. The string displayed as the name of the device group, which must be unique in the organization.
Serial Numbers	String	Required. The strings containing the serial numbers of the devices in the group.

Notes:

- All identities must be configured as both SSL clients and servers.
- Leader certificates must have the common name prefix `leader` (case insensitive).
- Member certificates must have the common name prefix `member` (case insensitive).
- Student configuration profiles are not supported on macOS and will be rejected at installation time.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

32

Apple ID	String	Specifying the specified resource.
Passcode Type	String	Optional. The Managed Apple ID for this user. Optional. The passcode UI to show when the user is at the login window; possible values are <code>com.apple.ersix</code>

The Device Group must contain an array of dictionaries with the following key-value pairs:

Key	Type	Content
Identifier	String	Required. The string that uniquely identifies the device group in the organization.
Name	String	Required. The string displayed as the name of the device group, which must be unique in the organization.
Serial Numbers	String	Required. The strings containing the serial numbers of the devices in the group.

Notes:

- All identities must be configured as both SSL clients and servers.
- Leader certificates must have the common name prefix `leader` (case insensitive).
- Member certificates must have the common name prefix `member` (case insensitive).

Email Payload

The email payload is designated by specifying `com.apple.mail` as the `Payload` 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
Email Account Description	String	Optional. A user-visible description of the email account, shown in the Mail and Settings applications.
Email Account Name	String	Optional. The full user name for the account. This is the user name in sent messages, etc.
Email Account Type	String	Allowed values are <code>EmailTypeIMAP</code> or <code>EmailTypeIMAP</code> . Defines the protocol to be used for that account.
Email Address	String	Designates the full email address for the account. If not present in the payload, the device prompts for this string during profile installation.
Incoming Mail Server Authentication	String	Designates the authentication scheme for incoming mail. Allowed values are <code>EmailAuthPAM</code> , <code>EmailAuthNTLM</code> , <code>EmailAuthTLS</code> , <code>EmailAuthNone</code> .
Incoming Mail Server Host Name	String	Designates the incoming mail server host name (or IP address).
Incoming Mail Server String	String	Optional. The path prefix for the IMAP mail server.
IMAP Path Prefix	String	
Incoming Mail Server Integer Port Number	Integer	Optional. Designates the incoming mail server port number. If no port number is specified, the default port for a given protocol is used.
Incoming Mail Server SSL	Boolean	Optional. Default false. Designates whether the incoming mail server uses SSL for authentication.
Incoming Mail Server String	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.
Incoming Password	String	Optional. Password for the Incoming Mail Server. Use only with encrypted profiles.
Outgoing Password	String	Optional. Password for the Outgoing Mail Server. Use only with encrypted profiles.
Outgoing Password Subfields	String	Optional. If set, the user will be prompted for the password only once and it will be used for both outgoing and incoming mail.
Outgoing Mail Server Authentication	String	Designates the authentication scheme for outgoing mail. Allowed values are <code>EmailAuthPAM</code> , <code>EmailAuthNTLM</code> , <code>EmailAuthTLS</code> , <code>EmailAuthNone</code> .

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

33

Email Payload

The email payload is designated by specifying `com.apple.mail` as the `Payload` 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
Email Account Description	String	Optional. A user-visible description of the email account, shown in the Mail and Settings applications.
Email Account Name	String	Optional. The full user name for the account. This is the user name in sent messages, etc.
Email Account Type	String	Allowed values are <code>EmailTypeIMAP</code> or <code>EmailTypeIMAP</code> . Defines the protocol to be used for that account.
Email Address	String	Designates the full email address for the account. If not present in the payload, the device prompts for this string during profile installation.
Incoming Mail Server Authentication	String	Designates the authentication scheme for incoming mail. Allowed values are <code>EmailAuthPAM</code> , <code>EmailAuthNTLM</code> , <code>EmailAuthTLS</code> , <code>EmailAuthNone</code> .
Incoming Mail Server Host Name	String	Designates the incoming mail server host name (or IP address).
Incoming Mail Server String	String	Optional. The path prefix for the IMAP mail server.
IMAP Path Prefix	String	
Incoming Mail Server Integer Port Number	Integer	Optional. Designates the incoming mail server port number. If no port number is specified, the default port for a given protocol is used.
Incoming Mail Server SSL	Boolean	Optional. Default false. Designates whether the incoming mail server uses SSL for authentication.
Incoming Mail Server String	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.
Incoming Password	String	Optional. Password for the Incoming Mail Server. Use only with encrypted profiles.
Outgoing Password	String	Optional. Password for the Outgoing Mail Server. Use only with encrypted profiles.
Outgoing Password Subfields	String	Optional. If set, the user will be prompted for the password only once and it will be used for both outgoing and incoming mail.
Outgoing Mail Server Authentication	String	Designates the authentication scheme for outgoing mail. Allowed values are <code>EmailAuthPAM</code> , <code>EmailAuthNTLM</code> , <code>EmailAuthTLS</code> , <code>EmailAuthNone</code> .

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

33

Key	Type	Value
Device Group	String	Designates the device group to which the profile belongs.

Key	Type	Value
Device Group	String	Designates the device group to which the profile belongs.

OutgoingMailServerHostname	String	Designates the outgoing mail server host name (or IP address).
OutgoingMailServerPortNumber	Integer	Optional. Designates the outgoing mail server port number. If no port number is specified, ports 25, 587 and 465 are used, in this order.
OutgoingMailServerSSL	Boolean	Optional. Default is false. Designates whether the outgoing mail server uses SSL for authentication.
OutgoingMailServerString	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 is 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 is false. If true, this account is not available for sending mail in any app other than the Apple Mail app.
Availability: Available only in iOS 5.0 and later.		
SMIMEEnabled	Boolean	Optional. Default is false. This account supports S/MIME. As of iOS 10.0, this key is ignored.
Availability: Available only in iOS 5.0 through iOS 9.3.3.		
SMIMESigningEnabled	Boolean	Optional. Default is false. If set to true, S/MIME signing is enabled for this account.
Availability: Available only in iOS 10.3 and later.		
SMIMESigningCertificateUUID	String	Optional. The Pay load of the Identity certificate used to sign messages sent from this account.
Availability: Available only in iOS 5.0 and later.		
SMIMEEncryptionEnabled	Boolean	Optional. Default is false. If set to true, S/MIME encryption is on by default for this account.
Availability: Available only in iOS 10.3 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEncryptionByDefault .		
SMIMEEncryptionCertificateUUID	String	Optional. The Pay load of the Identity certificate used to decrypt messages sent to this account. The public certificate is attached to outgoing mail to allow encrypted mail to be sent to this user. When the user sends encrypted mail, the public certificate is used to encrypt the copy of the mail in their Sent mailbox.
Availability: Available only in iOS 5.0 and later.		

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

34

Key	Type	Value
SMIMEEnablePerMessageSwitch	Boolean	Optional. Default is false. Displays the per-message encryption switch in the Mail Compose UI.
Availability: Available only in iOS 8.0 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEnableEncryptionPerMessageSwitch .		
disableMailRecentSwitch	Boolean	Optional. If true, this account is excluded from address Recents syncing. This defaults to false.
Availability: Available only in iOS 6.0 and later.		
allowMailDrop	Boolean	Optional. If true, this account is allowed to use Mail Drop. The default is false.
Availability: Available in iOS 9.2 and macOS 10.12 and later.		
SMIMESigningUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can toggle S/MIME signing on or off in Settings.
Availability: Available only in iOS 12.0 and later.		
SMIMESigningCertificateUUIDUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can select the signing identity.
Availability: Available only in iOS 12.0 and later.		
SMIMEEncryptByDefault	Boolean	Optional. Default is false. If set to true, S/MIME encryption is enabled by default. If SMIMEEnableEncryptionPerMessageSwitch is false, this default cannot be changed by the user.
Availability: Available only in iOS 12.0 and later.		
SMIMEEncryptByDefaultUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can toggle the encryption by default setting.
Availability: Available only in iOS 12.0 and later.		
SMIMEEncryptionCertificateUUIDUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can select the S/MIME encryption identity and encryption is enabled.
Availability: Available only in iOS 12.0 and later.		
SMIMEEnableEncryptionMessageSwitch	Boolean	Optional. Default is false. If set to true, display the per-message encryption switch in the Mail Compose UI.
Availability: Available only in iOS 12.0 and later.		

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

35

OutgoingMailServerHostname	String	Designates the outgoing mail server host name (or IP address).
OutgoingMailServerPortNumber	Integer	Optional. Designates the outgoing mail server port number. If no port number is specified, ports 25, 587 and 465 are used, in this order.
OutgoingMailServerSSL	Boolean	Optional. Default is false. Designates whether the outgoing mail server uses SSL for authentication.
OutgoingMailServerString	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 is 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 is false. If true, this account is not available for sending mail in any app other than the Apple Mail app.
Availability: Available only in iOS 5.0 and later.		
SMIMEEnabled	Boolean	Optional. Default is false. This account supports S/MIME. As of iOS 10.0, this key is ignored.
Availability: Available only in iOS 5.0 through iOS 9.3.3.		
SMIMESigningEnabled	Boolean	Optional. Default is false. If set to true, S/MIME signing is enabled for this account.
Availability: Available only in iOS 10.3 and later.		
SMIMESigningCertificateUUID	String	Optional. The Pay load of the Identity certificate used to sign messages sent from this account.
Availability: Available only in iOS 5.0 and later.		
SMIMEEncryptionEnabled	Boolean	Optional. Default is false. If set to true, S/MIME encryption is on by default for this account.
Availability: Available only in iOS 10.3 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEncryptionByDefault .		
SMIMEEncryptionCertificateUUID	String	Optional. The Pay load of the Identity certificate used to decrypt messages sent to this account. The public certificate is attached to outgoing mail to allow encrypted mail to be sent to this user. When the user sends encrypted mail, the public certificate is used to encrypt the copy of the mail in their Sent mailbox.
Availability: Available only in iOS 5.0 and later.		

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

34

Key	Type	Value
SMIMEEnablePerMessageSwitch	Boolean	Optional. Default is false. Displays the per-message encryption switch in the Mail Compose UI.
Availability: Available only in iOS 8.0 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEnableEncryptionPerMessageSwitch .		
disableMailRecentSwitch	Boolean	Optional. If true, this account is excluded from address Recents syncing. This default is false.
Availability: Available only in iOS 6.0 and later.		
allowMailDrop	Boolean	Optional. If true, this account is allowed to use Mail Drop. The default is false.
Availability: Available in iOS 9.2 and macOS 10.12 and later.		
SMIMESigningUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can toggle S/MIME signing on or off in Settings.
Availability: Available only in iOS 12.0 and later.		
SMIMESigningCertificateUUIDUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can select the signing identity.
Availability: Available only in iOS 12.0 and later.		
SMIMEEncryptByDefault	Boolean	Optional. Default is false. If set to true, S/MIME encryption is enabled by default. If SMIMEEnableEncryptionPerMessageSwitch is false, this default cannot be changed by the user.
Availability: Available only in iOS 12.0 and later.		
SMIMEEncryptByDefaultUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can toggle the encryption by default setting.
Availability: Available only in iOS 12.0 and later.		
SMIMESigningCertificateUUIDUserOverrideable	Boolean	Optional. Default is false. If set to true, the user can select the S/MIME encryption identity and encryption is enabled.
Availability: Available only in iOS 12.0 and later.		
SMIMEEnableEncryptionMessageSwitch	Boolean	Optional. Default is false. If set to true, display the per-message encryption switch in the Mail Compose UI.
Availability: Available only in iOS 12.0 and later.		

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

35

802.1x Ethernet Payload

The 802.1x Ethernet payload is designated by specifying one of the following as the `Payload Value`:

- `com.apple.firstactiveet[default]et.managed`
- `com.apple.firstethernet.managed`
- `com.apple.secondactiveethernet.managed`
- `com.apple.secondethernet.managed`
- `com.apple.thirdactiveethernet.managed`
- `com.apple.thirdethernet.managed`
- `com.apple.globalethernet.managed`

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

Key	Type	Value
<code>InterfaceString</code>	String	The <code>com.apple.globalethernet</code> payload uses the value <code>d</code> . Any Ethernet values for the other payloads are derived from their name; for example the <code>com.apple.firstethernet</code> value would be <code>aged</code> FirstEthernet

Payloads with "active" in their name apply to Ethernet interfaces that are working at the time of profile installation. If there is no active Ethernet interface working, the `com.apple.firstactiveet[payloadValue].managed` configures the interface with the highest service order priority.

Payloads without "active" in the name apply to Ethernet interfaces according to service order regardless of whether the interface is working or not.

There is currently no support for a BSD level specifier.

To specify an enterprise profile for a given 802.1x network, include the `EAPClientConfiguration` key in the payload, as described in [EAPClientConfiguration Dictionary](#).

802.1x Ethernet Payload

The 802.1x Ethernet payload is designated by specifying one of the following as the `Payload Value`:

- `com.apple.firstactiveet[default]et.managed`
- `com.apple.firstethernet.managed`
- `com.apple.secondactiveethernet.managed`
- `com.apple.secondethernet.managed`
- `com.apple.thirdactiveethernet.managed`
- `com.apple.thirdethernet.managed`
- `com.apple.globalethernet.managed`

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

Key	Type	Value
<code>InterfaceString</code>	String	The <code>com.apple.globalethernet</code> payload uses the value <code>d</code> . Any Ethernet values for the other payloads are derived from their name; for example the <code>com.apple.firstethernet</code> value would be <code>aged</code> FirstEthernet

Payloads with "active" in their name apply to Ethernet interfaces that are working at the time of profile installation. If there is no active Ethernet interface working, the `com.apple.firstactiveet[payloadValue].managed` configures the interface with the highest service order priority.

Payloads without "active" in the name apply to Ethernet interfaces according to service order regardless of whether the interface is working or not.

There is currently no support for a BSD level specifier.

To specify an enterprise profile for a given 802.1x network, include the `EAPClientConfiguration` key in the payload, as described in [EAPClientConfiguration Dictionary](#).

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

36

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

36

Exchange Payload

In iOS, the Exchange payload is designated by specifying `com.apple.eas[stbedPoaynto a d]Value`. This payload configures an Exchange Active Sync Contacts account on the device. Mail and Calendar are not configured using this payload on iOS.

In macOS, the Exchange payload is designated by specifying `com.apple.eews[stbedPoaynto a d]Value`. This payload will configure an Exchange Web Services account for Contacts, Mail, Notes, Reminders, and Calendar.

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

Key	Type	Value
Available in both iOS and macOS		
<code>EmailAddress</code>	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 macOS, this key is required.
<code>Host</code>	String	Specifies the Exchange server host name (or IP address). In macOS 10.11 and later, this key is optional.
<code>SSL</code>	Boolean	Optional. Default YES. Specifies whether the Exchange server uses SSL for authentication.
<code>UserName</code>	String	This string specifies the user name for this Exchange account. Required in macOS or non-interactive installations (like MDM on iOS).
<code>Password</code>	String	Optional. The password of the account. Use only with encrypted profiles.
<code>OAuth</code>	Boolean	Optional. Specifies whether the connection should use OAuth for authentication. If enabled, a password should not be specified. This defaults to <code>false</code> Availability: Available only in iOS 12.0 and macOS 10.14 and later.
Available in iOS only		
<code>Certificate</code>	NSData blob	Optional. For accounts that allow authentication via certificate, a p12 identity certificate in NSData blob format.
<code>CertificateName</code>	String	Optional. Specifies the name or description of the certificate.
<code>CertificatePassword</code>	String	Optional. The password necessary for the p12 identity certificate. Used with mandatory encryption of profiles.
<code>PreventMove</code>	Boolean	Optional. Default <code>false</code> If set to <code>true</code> , 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.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

37

Exchange Payload

In iOS, the Exchange payload is designated by specifying `com.apple.eas[stbedPoaynto a d]Value`. This payload configures an Exchange Active Sync Contacts account on the device. Mail and Calendar are not configured using this payload on iOS.

In macOS, the Exchange payload is designated by specifying `com.apple.eews[stbedPoaynto a d]Value`. This payload will configure an Exchange Web Services account for Contacts, Mail, Notes, Reminders, and Calendar.

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

Key	Type	Value
Available in both iOS and macOS		
<code>EmailAddress</code>	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 macOS, this key is required.
<code>Host</code>	String	Specifies the Exchange server host name (or IP address). In macOS 10.11 and later, this key is optional.
<code>SSL</code>	Boolean	Optional. Default YES. Specifies whether the Exchange server uses SSL for authentication.
<code>UserName</code>	String	This string specifies the user name for this Exchange account. Required in macOS or non-interactive installations (like MDM on iOS).
<code>Password</code>	String	Optional. The password of the account. Use only with encrypted profiles.
<code>OAuth</code>	Boolean	Optional. Specifies whether the connection should use OAuth for authentication. If enabled, a password should not be specified. This defaults to <code>false</code> Availability: Available only in iOS 12.0 and macOS 10.14 and later.
Available in iOS only		
<code>Certificate</code>	NSData blob	Optional. For accounts that allow authentication via certificate, a p12 identity certificate in NSData blob format.
<code>CertificateName</code>	String	Optional. Specifies the name or description of the certificate.
<code>CertificatePassword</code>	String	Optional. The password necessary for the p12 identity certificate. Used with mandatory encryption of profiles.
<code>PreventMove</code>	Boolean	Optional. Default <code>false</code> If set to <code>true</code> , 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.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

37

Key	Type	Value
PreventAppSheet	Boolean	Optional. Default false. If set to true, this account will not be available for sending mail in any app other than the Apple Mail app. Availability: Available in iOS 5.0 and later.
PayloadCertificateUUID	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. As of iOS 10.0, this key is ignored. Availability: Available only in iOS 5.0 through 9.3.3.
SMIMESigningEnabled	Boolean	Optional. Default true. If set to true, S/MIME signing is enabled for this account. Availability: Available only in iOS 10.3 and later.
SMIMESigningCertificateUUID	String	Optional. The payload of the identity certificate used to sign messages sent from this account. Availability: Available only in iOS 5.0 and later.
SMIMEEncryptionEnabled	Boolean	Optional. Default false. If set to true, S/MIME encryption is on by default for this account. Availability: Available only in iOS 10.3 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEncryptionEnabled instead.
SMIMEEncryptionCertificateUUID	String	Optional. The payload of the identity certificate used to decrypt messages sent to this account. The public certificate is attached to outgoing mail to allow encrypted mail to be sent to this user. When the user sends encrypted mail, the public certificate is used to encrypt the copy of the mail in their Sent mailbox. Availability: Available only in iOS 5.0 and later.
SMIMEEnablePerMessageSwitch	Boolean	Optional. Default false. If set to true, displays the per-message encryption switch in the Mail Compose UI. Availability: Available only in iOS 8.0 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEnableEncryptionPerMessageSwitch instead.
SMIMESigningUserOverrideable	Boolean	Optional. Default false. If set to true, the user can toggle S/MIME signing on or off in Settings. Availability: Available only in iOS 12.0 and later.
SMIMESigningCertificateUUID	String	Optional. Default false. If set to true, the user can select the signing identity. Availability: Available only in iOS 12.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

38

Key	Type	Value
PreventAppSheet	Boolean	Optional. Default false. If set to true, this account will not be available for sending mail in any app other than the Apple Mail app. Availability: Available in iOS 5.0 and later.
PayloadCertificateUUID	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. As of iOS 10.0, this key is ignored. Availability: Available only in iOS 5.0 through 9.3.3.
SMIMESigningEnabled	Boolean	Optional. Default true. If set to true, S/MIME signing is enabled for this account. Availability: Available only in iOS 10.3 and later.
SMIMESigningCertificateUUID	String	Optional. The payload of the identity certificate used to sign messages sent from this account. Availability: Available only in iOS 5.0 and later.
SMIMEEncryptionEnabled	Boolean	Optional. Default false. If set to true, S/MIME encryption is on by default for this account. Availability: Available only in iOS 10.3 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEncryptionEnabled instead.
SMIMEEncryptionCertificateUUID	String	Optional. The payload of the identity certificate used to decrypt messages sent to this account. The public certificate is attached to outgoing mail to allow encrypted mail to be sent to this user. When the user sends encrypted mail, the public certificate is used to encrypt the copy of the mail in their Sent mailbox. Availability: Available only in iOS 5.0 and later.
SMIMEEnablePerMessageSwitch	Boolean	Optional. Default false. If set to true, displays the per-message encryption switch in the Mail Compose UI. Availability: Available only in iOS 8.0 and later. As of iOS 12.0, this key is deprecated. It is recommended to use SMIMEEnableEncryptionPerMessageSwitch instead.
SMIMESigningUserOverrideable	Boolean	Optional. Default false. If set to true, the user can toggle S/MIME signing on or off in Settings. Availability: Available only in iOS 12.0 and later.
SMIMESigningCertificateUUID	String	Optional. Default false. If set to true, the user can select the signing identity. Availability: Available only in iOS 12.0 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

38

Key	Type	Value
SMIMEEncryptByDefault	Boolean	Optional. Default false. If S/MIME encryption is enabled by default, if SMIMEEnableEncryptionPerMessageSwitch is false, this default cannot be changed by the user. Availability: Available only in iOS 12.0 and later.
SMIMEEncryptByDefaultOverrides	Boolean	Optional. Default false. If set to true, the user can toggle the encryption by default setting. Availability: Available only in iOS 12.0 and later.
SMIMEEncryptionCertificateUUID	String	Optional. Default false. If set to true, the user can select the S/MIME encryption identity and encryption is enabled. Availability: Available only in iOS 12.0 and later.
SMIMEEnableEncryptMessageSwitch	Boolean	Optional. Default false. If set to true, displays the per-message encryption switch in the Mail Compose UI. Availability: Available only in iOS 12.0 and later.
allowMailDrop	Boolean	Optional. If true, this account is allowed to use Mail Drop. The default is false. Availability: Available only in macOS 10.12 and later.
disableMailRecents	Boolean	Optional. If true, this account is excluded from address Recents syncing. This defaults to false. Availability: Available only in iOS 6.0 and later.
MailNumberOfPastDaysSync	String	Optional. The number of days since synchronization. CommunicationServiceDictionary is optional. The communication service handler rules for this account. The CommunicationServiceRules dictionary currently contains only a DefaultServiceKey. Its value is a dictionary which contains an AudioKey whose value is a string containing the bundle identifier for the default application that handles audio calls made to contacts from this account. Availability: Available in macOS Only
Path	String	Optional.
Port	Integer	Optional.
ExternalHost	String	Optional.
ExternalSSL	Boolean	Optional.
ExternalPath	String	Optional.
ExternalPort	Integer	Optional.
OAuthSignInURL	String	Optional. Specifies the URL to load into a webview for authentication via OAuth when auto-discovery is not used. Requires a Host value. Availability: Available only in macOS 10.14 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

39

Key	Type	Value
SMIMEEncryptByDefault	Boolean	Optional. Default false. If S/MIME encryption is enabled by default, if SMIMEEnableEncryptionPerMessageSwitch is false, this default cannot be changed by the user. Availability: Available only in iOS 12.0 and later.
SMIMEEncryptByDefaultOverrides	Boolean	Optional. Default false. If set to true, the user can toggle the encryption by default setting. Availability: Available only in iOS 12.0 and later.
SMIMEEncryptionCertificateUUID	String	Optional. Default false. If set to true, the user can select the S/MIME encryption identity and encryption is enabled. Availability: Available only in iOS 12.0 and later.
SMIMEEnableEncryptMessageSwitch	Boolean	Optional. Default false. If set to true, displays the per-message encryption switch in the Mail Compose UI. Availability: Available only in iOS 12.0 and later.
allowMailDrop	Boolean	Optional. If true, this account is allowed to use Mail Drop. The default is false. Availability: Available only in macOS 10.12 and later.
disableMailRecents	Boolean	Optional. If true, this account is excluded from address Recents syncing. This defaults to false. Availability: Available only in iOS 6.0 and later.
MailNumberOfPastDaysSync	String	Optional. The number of days since synchronization. CommunicationServiceDictionary is optional. The communication service handler rules for this account. The CommunicationServiceRules dictionary currently contains only a DefaultServiceKey. Its value is a dictionary which contains an AudioKey whose value is a string containing the bundle identifier for the default application that handles audio calls made to contacts from this account. Availability: Available in macOS Only
Path	String	Optional.
Port	Integer	Optional.
ExternalHost	String	Optional.
ExternalSSL	Boolean	Optional.
ExternalPath	String	Optional.
ExternalPort	Integer	Optional.
OAuthSignInURL	String	Optional. Specifies the URL to load into a webview for authentication via OAuth when auto-discovery is not used. Requires a Host value. Availability: Available only in macOS 10.14 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

39

Note

As with VPN and Wi-Fi configurations, it is possible to associate an SCEP credential with an Exchange configuration via the `Pay load Certificate UUID`

Note

As with VPN and Wi-Fi configurations, it is possible to associate an SCEP credential with an Exchange configuration via the `Pay load Certificate UUID`

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

40

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

40

FileVault 2

In macOS 10.9, you can use FileVault 2 to perform full XTS-AES 128 encryption on the contents of a volume. FileVault 2 payloads are designated by specifying `com.apple.MCX.Fi` as the `Pay load`. Removal of the FileVault payload does not disable FileVault.

Key	Type	Value
<code>Enable</code>	String	Set to 'On' to enable FileVault. Set to 'Off' to disable FileVault. This value is required.
<code>Defer</code>	Boolean	Set to <code>true</code> defer enabling FileVault until the designated user logs out. For details, see <code>fdesetup(8)</code> . The person enabling FileVault must be either a local user or a mobile account user.
<code>UserEnterMissingProfile</code>	Boolean	Set to <code>true</code> manual profile installs to prompt for missing user name or password fields.
<code>UserRecoveryKey</code>	Boolean	Set to <code>true</code> create a personal recovery key. Defaults to <code>true</code>
<code>ShowRecoveryKey</code>	Boolean	Set to <code>false</code> not display the personal recovery key to the user after FileVault is enabled. Defaults to <code>true</code>
<code>OutputPath</code>	String	Path to the location where the recovery key and computer information plist will be stored.
<code>Certificate</code>	Data	DER-encoded certificate data if an institutional recovery key will be added.
<code>Pay load Certificate UUID</code>	UUID	UUID of the payload containing the asymmetric recovery key certificate payload.
<code>Username</code>	String	User name of the Open Directory user that will be added to FileVault.
<code>Password</code>	String	User password of the Open Directory user that will be added to FileVault. Use the <code>UserEnterMissingKey</code> if you want to prompt for this information.
<code>UseKeychain</code>	Boolean	If set to <code>true</code> and no certificate information is provided in this payload, the keychain already created at <code>/Library/Keychains/FileVaultMaster.keychain</code> will be used when the institutional recovery key is added.
<code>DeferForceAtUseAttempts</code>	Integer	When using the <code>Defer</code> option you can optionally set this key to the maximum number of times the user can bypass enabling FileVault before it will require that it be enabled before the user can log in. If set to 0, it will always prompt to enable FileVault until it is enabled, though it will allow you to bypass enabling it. Setting this key to -1 will disable this feature. Availability: Available in macOS 10.10 and later.
<code>DeferDontAskAtLogout</code>	Boolean	When using the <code>Defer</code> option, set this key to <code>true</code> not request enabling FileVault at user logout time. Availability: Available in macOS 10.10 and later.

A personal recovery user will normally be created unless the `UserRecoveryKey` is `false`. An institutional recovery key will be created only if either there is certificate data available in the `Certificate` value, a specific certificate payload is referenced, or the `UseKeychain` is set to `true` and a valid

FileVault 2

In macOS 10.9, you can use FileVault 2 to perform full XTS-AES 128 encryption on the contents of a volume. FileVault 2 payloads are designated by specifying `com.apple.MCX.Fi` as the `Pay load`. Removal of the FileVault payload does not disable FileVault.

Key	Type	Value
<code>Enable</code>	String	Set to 'On' to enable FileVault. Set to 'Off' to disable FileVault. This value is required.
<code>Defer</code>	Boolean	Set to <code>true</code> defer enabling FileVault until the designated user logs out. For details, see <code>fdesetup(8)</code> . The person enabling FileVault must be either a local user or a mobile account user.
<code>UserEnterMissingProfile</code>	Boolean	Set to <code>true</code> manual profile installs to prompt for missing user name or password fields.
<code>UserRecoveryKey</code>	Boolean	Set to <code>true</code> create a personal recovery key. Defaults to <code>true</code>
<code>ShowRecoveryKey</code>	Boolean	Set to <code>false</code> not display the personal recovery key to the user after FileVault is enabled. Defaults to <code>true</code>
<code>OutputPath</code>	String	Path to the location where the recovery key and computer information plist will be stored.
<code>Certificate</code>	Data	DER-encoded certificate data if an institutional recovery key will be added.
<code>Pay load Certificate UUID</code>	UUID	UUID of the payload containing the asymmetric recovery key certificate payload.
<code>Username</code>	String	User name of the Open Directory user that will be added to FileVault.
<code>Password</code>	String	User password of the Open Directory user that will be added to FileVault. Use the <code>UserEnterMissingKey</code> if you want to prompt for this information.
<code>UseKeychain</code>	Boolean	If set to <code>true</code> and no certificate information is provided in this payload, the keychain already created at <code>/Library/Keychains/FileVaultMaster.keychain</code> will be used when the institutional recovery key is added.
<code>DeferForceAtUseAttempts</code>	Integer	When using the <code>Defer</code> option you can optionally set this key to the maximum number of times the user can bypass enabling FileVault before it will require that it be enabled before the user can log in. If set to 0, it will always prompt to enable FileVault until it is enabled, though it will allow you to bypass enabling it. Setting this key to -1 will disable this feature. Availability: Available in macOS 10.10 and later.
<code>DeferDontAskAtLogout</code>	Boolean	When using the <code>Defer</code> option, set this key to <code>true</code> not request enabling FileVault at user logout time. Availability: Available in macOS 10.10 and later.

A personal recovery user will normally be created unless the `UserRecoveryKey` is `false`. An institutional recovery key will be created only if either there is certificate data available in the `Certificate` value, a specific certificate payload is referenced, or the `UseKeychain` is set to `true` and a valid

FileVault Master file key created: In all cases, the certificate information must be set up properly for FileVault or it will be ignored and no institutional recovery key will be set up.

FDE Recovery Key Escrow Payload

FileVault Full Disk Encryption (FDE) recovery keys are, by default, sent to Apple if the user requests it. Starting with macOS 10.13, recovery key escrow payloads are designated by specifying `com.apple.security.FDERecovery as the PayLoadType`. Only one payload of this type is allowed per system.

If FileVault is enabled after this payload is installed on the system, the FileVault PRK will be encrypted with the specified certificate, wrapped with a CMS envelope and stored at `/var/db/FileVault/TheEncryptedData`. The `TheEncryptedData` will be made available to the MDM server as part of the `Security` payload. Alternatively, if a site uses its own administration software, it can extract the PRK from the foregoing location at any time. Because the PRK is encrypted using the certificate provided in the profile, only the author of the profile can extract the data.

Note these cautions:

- The payload must exist in a system-scoped profile.
- Installing more than one payload of this type per machine will cause an error.
- The previous payload (`com.apple.security.FDERecovery as the PayLoadType`) is no longer supported; it can still be installed, but it will be ignored. This lets servers send out the same profile to old and new clients.
- If only an old-style redirection payload is installed at the time FileVault is turned on (by means of the Security Preferences pane), an error will be displayed and FileVault will not be enabled.
- No warning or error will be provided if FileVault is already enabled and an old-style payload is installed. In this case, it's assumed that the recovery key has already been escrowed with the server.

This payload contains these keys:

Key	Type	Value
<code>Location</code>	String	Required. A short description of the location where the recovery key will be escrowed. This text will be inserted into the message the user sees when enabling FileVault.
<code>EncryptionCertificatePayloadUUID</code>	String	Required. The UUID of a payload within the same profile that contains the certificate that will be used to encrypt the recovery key. The referenced payload must be of type <code>com.apple.security.pkcs1</code> .
<code>DeviceKey</code>	String	Optional. An optional string that will be included in help text if the user appears to have forgotten the password. Can be used by a site admin to look up the escrowed key for the particular machine. Replaces the <code>RecordNumber</code> used in previous escrow mechanism. If missing, the device serial number will be used instead.

Although the previous FDE Recovery payload is no longer supported on macOS 10.13 and later, it is still supported on macOS 10.9 through 10.12. This payload is designated by specifying `com.apple.security.FDERecovery as the PayLoadType`.

Once installed, this payload will cause any FDE (Full Disk Encryption) 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.

Note these cautions:

- The payload must exist in a system-scoped profile.
- Installing more than one payload of this type per machine will cause an error.

This payload contains these keys:

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

FileVault Client Request

The client issues a HTTPS POST request to the server with XML data containing the following:

Key	Type	Value
<code>VersionNumber</code>	String	Currently set to '1.0'.
<code>SerialNumber</code>	String	The serial number of the client computer. The server must include this value in its response back to the client (see below).
<code>RecoveryKey</code>	String	4

The recovery key encrypted using the encryption certificate provided in the configuration profile (referenced by the `EncryptionCertificatePayloadUUID`). The `RecoveryKey` payload contains only the recovery key string without any XML wrapper. The encrypted data is wrapped in a CMS envelope and is then Base-64 encoded.

These tags are enclosed within a parent `FDECaptureRequest`. An example of an XML message body is:

```
<FDECaptureRequest>
<VersionNumber>1.0</VersionNumber>
<SerialNumber>A02FE08UCC8X</SerialNumber>
<RecoveryKey>CMS64>MIAGCSqGSIb3DQEHA ... AAAA AAAAAA = </RecoveryKey>
</FDECaptureRequest>
```

FileVault Master file key created: In all cases, the certificate information must be set up properly for FileVault or it will be ignored and no institutional recovery key will be set up.

FDE Recovery Key Escrow Payload

FileVault Full Disk Encryption (FDE) recovery keys are, by default, sent to Apple if the user requests it. Starting with macOS 10.13, recovery key escrow payloads are designated by specifying `com.apple.security.FDERecovery as the PayLoadType`. Only one payload of this type is allowed per system.

If FileVault is enabled after this payload is installed on the system, the FileVault PRK will be encrypted with the specified certificate, wrapped with a CMS envelope and stored at `/var/db/FileVault/TheEncryptedData`. The `TheEncryptedData` will be made available to the MDM server as part of the `Security` payload. Alternatively, if a site uses its own administration software, it can extract the PRK from the foregoing location at any time. Because the PRK is encrypted using the certificate provided in the profile, only the author of the profile can extract the data.

Note these cautions:

- The payload must exist in a system-scoped profile.
- Installing more than one payload of this type per machine will cause an error.
- The previous payload (`com.apple.security.FDERecovery as the PayLoadType`) is no longer supported; it can still be installed, but it will be ignored. This lets servers send out the same profile to old and new clients.
- If only an old-style redirection payload is installed at the time FileVault is turned on (by means of the Security Preferences pane), an error will be displayed and FileVault will not be enabled.
- No warning or error will be provided if FileVault is already enabled and an old-style payload is installed. In this case, it's assumed that the recovery key has already been escrowed with the server.

This payload contains these keys:

Key	Type	Value
<code>Location</code>	String	Required. A short description of the location where the recovery key will be escrowed. This text will be inserted into the message the user sees when enabling FileVault.
<code>EncryptionCertificatePayloadUUID</code>	String	Required. The UUID of a payload within the same profile that contains the certificate that will be used to encrypt the recovery key. The referenced payload must be of type <code>com.apple.security.pkcs1</code> .
<code>DeviceKey</code>	String	Optional. An optional string that will be included in help text if the user appears to have forgotten the password. Can be used by a site admin to look up the escrowed key for the particular machine. Replaces the <code>RecordNumber</code> used in previous escrow mechanism. If missing, the device serial number will be used instead.

Although the previous FDE Recovery payload is no longer supported on macOS 10.13 and later, it is still supported on macOS 10.9 through 10.12. This payload is designated by specifying `com.apple.security.FDERecovery as the PayLoadType`.

Once installed, this payload will cause any FDE (Full Disk Encryption) 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.

Note these cautions:

- The payload must exist in a system-scoped profile.
- Installing more than one payload of this type per machine will cause an error.

This payload contains these keys:

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

FileVault Client Request

The client issues a HTTPS POST request to the server with XML data containing the following:

Key	Type	Value
<code>VersionNumber</code>	String	Currently set to '1.0'.
<code>SerialNumber</code>	String	The serial number of the client computer. The server must include this value in its response back to the client (see below).
<code>RecoveryKey</code>	String	4

The recovery key encrypted using the encryption certificate provided in the configuration profile (referenced by the `EncryptionCertificatePayloadUUID`). The `RecoveryKey` payload contains only the recovery key string without any XML wrapper. The encrypted data is wrapped in a CMS envelope and is then Base-64 encoded.

These tags are enclosed within a parent `FDECaptureRequest`. An example of an XML message body is:

```
<FDECaptureRequest>
<VersionNumber>1.0</VersionNumber>
<SerialNumber>A02FE08UCC8X</SerialNumber>
<RecoveryKey>CMS64>MIAGCSqGSIb3DQEHA ... AAAA AAAAAA = </RecoveryKey>
</FDECaptureRequest>
```

FileVault Server Response

Upon receiving the client's request, the server must respond to the client with XML data containing:

Key	Type	Value
<code>SerialNumber</code> <code>String</code>		The serial number of the client computer. This value must be the same as the one sent in the request.
<code>RecordNumber</code> <code>String</code>		This value must be nonempty but otherwise is up to the site to define it. This value will be displayed to the user along with the serial number on the EFI login screen when the user is asked to enter the recovery key. As an example, this could be a value to assist the site administrator in locating or verifying the user's recovery key in a database.

Firewall Payload

Available in macOS 10.12 and later. A Firewall payload manages the Application Firewall settings accessible in the Security Preferences pane. Note these restrictions:

- The payload must exist in a system-scoped profile.
- If more than one profile contains this payload, the most restrictive union of settings will be used.
- The "Automatically allow signed downloaded software" and "Automatically allow built-in software" options are not supported, but both will be forced ON when this payload is present.

This payload is designated by specifying `com.apple.security.firewall` as `Value`.

The Firewall payload contains the following keys:

Key	Type	Value
<code>EnableFirewall</code> <code>Boolean</code>		Required. Whether the firewall should be enabled or not.
<code>BlockAllIncomings</code> <code>Boolean</code>		Optional. Corresponds to the "Block all incoming connections" option.
<code>EnableStealthMode</code> <code>Boolean</code>		Optional. Corresponds to "Enable stealth mode."
<code>Applications</code> <code>Array of Dictionaries</code>		Optional. The list of applications. Each dictionary contains these keys: <ul style="list-style-type: none"> <code>BundleIdentifier</code> : identifies the application <code>Allows</code> <code>Boolean</code> : specifies whether or not incoming connections are allowed

Font Payload

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

A Font payload contains the following keys:

Key	Type	Value
<code>Name</code> <code>String</code>		Optional. The user-visible name for the font. This field is replaced by the actual name of the font after installation.
<code>FontData</code>		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.

Note

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.hasTheProxyBoardType`

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
<code>ProxyType</code> <code>String</code>		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.
<code>ProxyServer</code> <code>String</code>		The proxy server's network address.
<code>ProxyServerPort</code> <code>Integer</code>		The proxy server's port.
<code>ProxyUsername</code> <code>String</code>		Optional. The username used to authenticate to the proxy server.
<code>ProxyPassword</code> <code>String</code>		Optional. The password used to authenticate to the proxy server.

FileVault Server Response

Upon receiving the client's request, the server must respond to the client with XML data containing:

Key	Type	Value
<code>SerialNumber</code> <code>String</code>		The serial number of the client computer. This value must be the same as the one sent in the request.
<code>RecordNumber</code> <code>String</code>		This value must be nonempty but otherwise is up to the site to define it. This value will be displayed to the user along with the serial number on the EFI login screen when the user is asked to enter the recovery key. As an example, this could be a value to assist the site administrator in locating or verifying the user's recovery key in a database.

Firewall Payload

Available in macOS 10.12 and later. A Firewall payload manages the Application Firewall settings accessible in the Security Preferences pane. Note these restrictions:

- The payload must exist in a system-scoped profile.
- If more than one profile contains this payload, the most restrictive union of settings will be used.
- The "Automatically allow signed downloaded software" and "Automatically allow built-in software" options are not supported, but both will be forced ON when this payload is present.

This payload is designated by specifying `com.apple.security.firewall` as `Value`.

The Firewall payload contains the following keys:

Key	Type	Value
<code>EnableFirewall</code> <code>Boolean</code>		Required. Whether the firewall should be enabled or not.
<code>BlockAllIncomings</code> <code>Boolean</code>		Optional. Corresponds to the "Block all incoming connections" option.
<code>EnableStealthMode</code> <code>Boolean</code>		Optional. Corresponds to "Enable stealth mode."
<code>Applications</code> <code>Array of Dictionaries</code>		Optional. The list of applications. Each dictionary contains these keys: <ul style="list-style-type: none"> <code>BundleIdentifier</code> : identifies the application <code>Allows</code> <code>Boolean</code> : specifies whether or not incoming connections are allowed

Font Payload

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

A Font payload contains the following keys:

Key	Type	Value
<code>Name</code> <code>String</code>		Optional. The user-visible name for the font. This field is replaced by the actual name of the font after installation.
<code>FontData</code>		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.

Note

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.hasTheProxyBoardType`

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
<code>ProxyType</code> <code>String</code>		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.
<code>ProxyServer</code> <code>String</code>		The proxy server's network address.
<code>ProxyServerPort</code> <code>Integer</code>		The proxy server's port.
<code>ProxyUsername</code> <code>String</code>		Optional. The username used to authenticate to the proxy server.
<code>ProxyPassword</code> <code>String</code>		Optional. The password used to authenticate to the proxy server.

Proxy PAC URL	String	Optional. The URL of the PAC file that defines the proxy configuration.
---------------	--------	---

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

45

Proxy PAC URL	String	Optional. The URL of the PAC file that defines the proxy configuration.
---------------	--------	---

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

45

Key	Type	Value
Proxy PACFallbackBypass	Optional. If false prevents the device from connecting directly to the destination if the PAC file is unreachable. Default is false	
Availability:	Available in iOS 7 and later.	

Key	Type	Value
Proxy PACFallbackBypass	Optional. If false prevents the device from connecting directly to the destination if the PAC file is unreachable. Default is false	
Availability:	Available in iOS 7 and later.	

If the Proxy T field is set to A and no Proxy PAC URL is specified, the device uses the web proxy auto-discovery protocol (WPAD) to discover proxies.

Global Preferences Payload

The Global Preferences payload is designated by specifying .GlobalPreferencesPayload and Value. In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
MultipleSessionEnabled	Optional. If set to false user switching is disabled. Defaults to true	
com.apple.autologoutDelay	Optional. The auto log out delay (in seconds). A value of zero, means auto-logout is off. In some cases, this value may be restricted to values between 5 minutes and 24 hours.	

Google Account Payload

The Google account payload is designated by specifying com.apple.googleBuddy and Value. You can install multiple Google payloads.

Each Google payload sets up a Google email address as well as any other Google services the user enables after authentication. Google accounts must be installed via MDM or by Apple Configurator 2 (if the device is supervised). The payload never contains credentials; the user will be prompted to enter credentials shortly after the payload has been successfully installed.

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

Key	Type	Value
AccountDescription	Optional. A user-visible description of the Google account, shown in the Mail and Settings apps.	
Availability:	Available in iOS 9.3 and later.	
AccountName	Optional. The user's full name for the Google account. This name will appear in sent messages.	
Availability:	Available in iOS 9.3 and later.	

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

46

Proxy PAC URL	String	Optional. The URL of the PAC file that defines the proxy configuration.
---------------	--------	---

If the Proxy T field is set to A and no Proxy PAC URL is specified, the device uses the web proxy auto-discovery protocol (WPAD) to discover proxies.

Global Preferences Payload

The Global Preferences payload is designated by specifying .GlobalPreferencesPayload and Value. In addition to the settings common to all payloads, this payload defines the following keys:

Key	Type	Value
MultipleSessionEnabled	Optional. If set to false user switching is disabled. Defaults to true	
com.apple.autologoutDelay	Optional. The auto log out delay (in seconds). A value of zero, means auto-logout is off. In some cases, this value may be restricted to values between 5 minutes and 24 hours.	

Google Account Payload

The Google account payload is designated by specifying com.apple.googleBuddy and Value. You can install multiple Google payloads.

Each Google payload sets up a Google email address as well as any other Google services the user enables after authentication. Google accounts must be installed via MDM or by Apple Configurator 2 (if the device is supervised). The payload never contains credentials; the user will be prompted to enter credentials shortly after the payload has been successfully installed.

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

Key	Type	Value
AccountDescription	Optional. A user-visible description of the Google account, shown in the Mail and Settings apps.	
Availability:	Available in iOS 9.3 and later.	
AccountName	Optional. The user's full name for the Google account. This name will appear in sent messages.	
Availability:	Available in iOS 9.3 and later.	

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

46

Key	Type	Value
EmailAddress	String	Required. The full Google email address for the account.
Availability:	Available in iOS 9.3 and later.	
CommunicationServiceDictionary	Optional. The communication service handler rules for this account. The CommunicationServiceDictionaryRules currently contains only a DefaultServiceKey, whose value is a dictionary which contains an AudioKey, whose value is a string containing the bundle identifier for the default application that handles audio calls made to contacts from this account.	
Availability:	Available in iOS 10 and later.	

Key	Type	Value
EmailAddress	String	Required. The full Google email address for the account.
Availability:	Available in iOS 9.3 and later.	
CommunicationServiceDictionary	Optional. The communication service handler rules for this account. The CommunicationServiceDictionaryRules currently contains only a DefaultServiceKey, whose value is a dictionary which contains an AudioKey, whose value is a string containing the bundle identifier for the default application that handles audio calls made to contacts from this account.	
Availability:	Available in iOS 10 and later.	

Home Screen Layout Payload

The Home Screen Layout Payload is designated by specifying `com.apple.homeScreenLayout`. The payload can contain only one payload, which must be supervised. It is supported on the User Channel.

This payload defines a layout of apps, folders, and web clips for the Home screen. It is supported on iOS 9.3 and later.

Note

If a home screen layout puts more than six items in the iPad dock the location of the seventh and succeeding items may be undefined but they will not be omitted.

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

Key	Type	Value
<code>Dock Array</code>	Optional. An array of dictionaries, each of which must conform to the icon dictionary format. If it is not present, the user's dock will be empty.	
<code>Page Array</code>	Required. An array of arrays of dictionaries, each of which must conform to the icon dictionary format.	

Icon format dictionaries are defined as follows:

Key	Type	Value
<code>Type</code>	String	Required. Must be one of the following: <ul style="list-style-type: none"> Application Folder Web Clip
<code>DisplayName</code>	String	Optional. Human-readable string to be shown to the user. Valid only if <code>Folder</code> type.
<code>BundleID</code>	String	Required if <code>Application</code> . The bundle identifier of the app.
<code>Pages</code>	Array	Optional. An array of arrays of dictionaries, each of which must conform to the icon dictionary format. Valid only if <code>Folder</code> type.
<code>URL</code>	String	Required if <code>Web Clip</code> . URL of the WebClip being referenced. If more than one WebClip exists with the same URL, the behavior is undefined.
Availability: Available in iOS 11.3 and later.		

Identification Payload

The Identification payload is designated by specifying `com.apple.configurationprofile.ident` value as the `Payload` 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
<code>FullName</code>	String	The full name of the designated accounts.
<code>EmailAddress</code>	String	The address for the accounts.
<code>UserName</code>	String	The UNIX user name for the accounts.
<code>Password</code>	String	You can provide the password or choose to have the user provide it when he or she installs the profile.
<code>Prompt</code>	String	Custom instruction for the user, if needed.

Identity Preference Payload

Available in macOS 10.12 and later. An Identity Preference payload lets you identify an Identity Preference item in the user's keychain that references a Identity payload included in the same profile. It can only appear in a user profile, not a device profile. See also [Certificate Preference Payload](#) for setting up certificate preferences.

You can include multiple Identity Preference payloads as needed. Identity Preference payloads are designated by specifying `com.apple.security.identityPayload`.

An Identity Preference payload contains the following keys:

Key	Type	Value
<code>Name</code>	String	Required. An email address (RFC822), DNS hostname, or other name that uniquely identifies a service requiring this identity.
<code>PayloadUUID</code>	String	The UUID of another payload within the same profile that installed the identity; for example, a 'com.apple.security.pkcs12' or 'com.apple.security.scep' payload.

Home Screen Layout Payload

The Home Screen Layout Payload is designated by specifying `com.apple.homeScreenLayout`. The payload can contain only one payload, which must be supervised. It is supported on the User Channel.

This payload defines a layout of apps, folders, and web clips for the Home screen. It is supported on iOS 9.3 and later.

Note

If a home screen layout puts more than six items in the iPad dock the location of the seventh and succeeding items may be undefined but they will not be omitted.

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

Key	Type	Value
<code>Dock Array</code>	Optional. An array of dictionaries, each of which must conform to the icon dictionary format. If it is not present, the user's dock will be empty.	
<code>Page Array</code>	Required. An array of arrays of dictionaries, each of which must conform to the icon dictionary format.	

Icon format dictionaries are defined as follows:

Key	Type	Value
<code>Type</code>	String	Required. Must be one of the following: <ul style="list-style-type: none"> Application Folder Web Clip
<code>DisplayName</code>	String	Optional. Human-readable string to be shown to the user. Valid only if <code>Folder</code> type.
<code>BundleID</code>	String	Required if <code>Application</code> . The bundle identifier of the app.
<code>Pages</code>	Array	Optional. An array of arrays of dictionaries, each of which must conform to the icon dictionary format. Valid only if <code>Folder</code> type.
<code>URL</code>	String	Required if <code>Web Clip</code> . URL of the WebClip being referenced. If more than one WebClip exists with the same URL, the behavior is undefined.
Availability: Available in iOS 11.3 and later.		

Identification Payload

The Identification payload is designated by specifying `com.apple.configurationprofile.ident` value as the `Payload` 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
<code>FullName</code>	String	The full name of the designated accounts.
<code>EmailAddress</code>	String	The address for the accounts.
<code>UserName</code>	String	The UNIX user name for the accounts.
<code>Password</code>	String	You can provide the password or choose to have the user provide it when he or she installs the profile.
<code>Prompt</code>	String	Custom instruction for the user, if needed.

Identity Preference Payload

Available in macOS 10.12 and later. An Identity Preference payload lets you identify an Identity Preference item in the user's keychain that references a Identity payload included in the same profile. It can only appear in a user profile, not a device profile. See also [Certificate Preference Payload](#) for setting up certificate preferences.

You can include multiple Identity Preference payloads as needed. Identity Preference payloads are designated by specifying `com.apple.security.identityPayload`.

An Identity Preference payload contains the following keys:

Key	Type	Value
<code>Name</code>	String	Required. An email address (RFC822), DNS hostname, or other name that uniquely identifies a service requiring this identity.
<code>PayloadUUID</code>	String	The UUID of another payload within the same profile that installed the identity; for example, a 'com.apple.security.pkcs12' or 'com.apple.security.scep' payload.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

49

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

49

Kernel Extension Policy

The Kernel Extension Policy payload is designated by specifying `com.apple.syspolicy.kernel-extension` as the `payload` `value`. It is supported on macOS 10.13.2 and later.

This profile must be delivered via a user approved MDM server.

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

Key	Type	Value
<code>AllowUserOverride</code>	Boolean	If set to <code>true</code> , users can approve additional kernel extensions not explicitly allowed by configuration profiles.
<code>AllowedTeamIdentifiers</code>	Array	<code>value</code> is an array of team identifiers that define which validly signed kernel extensions will be allowed to load.
<code>AllowedKernelExtensions</code>	Dictionary	<code>value</code> is a dictionary representing a set of kernel extensions that will always be allowed to load on the machine. The dictionary maps team identifiers (keys) to arrays of bundle identifiers.

Kernel Extension Policy

The Kernel Extension Policy payload is designated by specifying `com.apple.syspolicy.kernel-extension` as the `payload` `value`. It is supported on macOS 10.13.2 and later.

This profile must be delivered via a user approved MDM server.

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

Key	Type	Value
<code>AllowUserOverride</code>	Boolean	If set to <code>true</code> , users can approve additional kernel extensions not explicitly allowed by configuration profiles.
<code>AllowedTeamIdentifiers</code>	Array	<code>value</code> is an array of team identifiers that define which validly signed kernel extensions will be allowed to load.
<code>AllowedKernelExtensions</code>	Dictionary	<code>value</code> is a dictionary representing a set of kernel extensions that will always be allowed to load on the machine. The dictionary maps team identifiers (keys) to arrays of bundle identifiers.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

50

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

50

LDAP Payload

The LDAP payload is designated by specifying `com.apple.ldap-as-a-keyboard-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
<code>LDAPAccountDescription</code>	String	Optional. Description of the account.
<code>LDAPAccountHostName</code>	String	The host.
<code>LDAPAccountUseSSL</code>	Boolean	Whether or not to use SSL.
<code>LDAPAccountUserName</code>	String	Optional. The username.
<code>LDAPAccountPassword</code>	String	Optional. Use only with encrypted profiles.
<code>LDAPSearchSettings</code>	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 <code>LDAPSearchSettings</code> 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).
<code>LDAPSearchSettingDescription</code>	String	Optional. Description of this search setting.
<code>LDAPSearchSettingScope</code>	String	Conceptually, the path to the node where a search should start. For example: <code>ou=people,o=example corp</code>
<code>LDAPSearchSettingRecursion</code>	String	Defines what recursion to use in the search. Can be one of the following 3 values: <ul style="list-style-type: none">• <code>LDAPSearchSettingScopeBase</code>: Just the immediate node pointed to by <code>SearchBase</code>• <code>LDAPSearchSettingScopeOneLevel</code>: The node plus its immediate children.• <code>LDAPSearchSettingScopeSubtree</code>: The node plus all children, regardless of depth.

LDAP Payload

The LDAP payload is designated by specifying `com.apple.ldap-as-a-keyboard-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
<code>LDAPAccountDescription</code>	String	Optional. Description of the account.
<code>LDAPAccountHostName</code>	String	The host.
<code>LDAPAccountUseSSL</code>	Boolean	Whether or not to use SSL.
<code>LDAPAccountUserName</code>	String	Optional. The username.
<code>LDAPAccountPassword</code>	String	Optional. Use only with encrypted profiles.
<code>LDAPSearchSettings</code>	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 <code>LDAPSearchSettings</code> 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).
<code>LDAPSearchSettingDescription</code>	String	Optional. Description of this search setting.
<code>LDAPSearchSettingScope</code>	String	Conceptually, the path to the node where a search should start. For example: <code>ou=people,o=example corp</code>
<code>LDAPSearchSettingRecursion</code>	String	Defines what recursion to use in the search. Can be one of the following 3 values: <ul style="list-style-type: none">• <code>LDAPSearchSettingScopeBase</code>: Just the immediate node pointed to by <code>SearchBase</code>• <code>LDAPSearchSettingScopeOneLevel</code>: The node plus its immediate children.• <code>LDAPSearchSettingScopeSubtree</code>: The node plus all children, regardless of depth.

Login Items Payload

The Login Items payload is designated by specifying `com.apple.loginitems.thePayloadType` value.

This payload handles login items on macOS. In versions previous to macOS 10.13, this information was in the `loginwindow`.

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

Key	Type	Value
<code>AutoLaunchedApplicationDict</code> Array of login item dictionaries.		

The Login Item dictionary contains:

Key	Type	Value
<code>path</code>	String	A URL or path string to the item's location.
<code>hide</code>	Boolean	Optional. If set to true, the item will not show up in the "Users & Groups" login items list.

Loginwindow Payload

The Loginwindow payload is designated by specifying `com.apple.loginwindow` and `value`. This payload creates managed preferences on all versions of macOS for system and device profiles. Multiple Loginwindow payloads may be installed together.

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

Key	Type	Value
<code>SHOWFULLNAME</code>	Boolean	Optional. Set to true to show the name and password dialog. Set to false to display a list of users.
<code>HideLocalUsers</code>	Boolean	Optional. When showing a user list, set to true to show only network and system users.
<code>IncludeNetworkUser</code>	Boolean	Optional. When showing a user list, set to true to show network users.
<code>HideAdminUsers</code>	Boolean	Optional. When showing a user list, set to false to hide the administrator users.
<code>SHOWOTHERUSERS_MANA</code>	Boolean	Optional. When showing a list of users, set to true to display Other... users.

Key	Type	Value
<code>AdminHostInfo</code>	String	Optional. If this key is included in the payload, its value will be displayed as additional computer information on the login window. Before macOS 10.10, this string could contain only particular information (Host Name, System Version, and so on). After macOS 10.10, setting this key to any value will allow the user to click the "time" area of the menu bar to toggle through various computer information values.
<code>AllowList</code>	Array of Strings	Optional. User or group GUIDs of users that are allowed to log in. An asterisk "*" string specifies all users or groups.
<code>DenyList</code>	Array of Strings	Optional. User or group GUIDs of users that cannot log in. This list takes priority over the list in the AllowList.
<code>HideMobileAccounts</code>	Boolean	Optional. If set to true, mobile account users will not be visible in a user list. In some cases mobile users will show up as network users.
<code>ShutdownDisabled</code>	Boolean	Optional. If set to true, the Shut Down button item will be hidden.
<code>RestartDisabled</code>	Boolean	Optional. If set to true, the Restart item will be hidden.
<code>SleepDisabled</code>	Boolean	Optional. If set to true, the Sleep button item will be hidden.
<code>DisableConsoleAccess</code>	Boolean	Optional. If set to true, the Other user will disregard use of the '>console' special user name.
<code>LoginwindowText</code>	String	Optional. Text to display in the login window.
<code>ShutdownDisabledWhileLoggedIn</code>	Boolean	Optional. If set to true, the Shut Down menu item will be disabled when the user is logged in.
<code>RestartDisabledWhileLoggedIn</code>	Boolean	Optional. If set to true, the Restart menu item will be disabled when the user is logged in.
<code>PowerOffDisabledWhileLoggedIn</code>	Boolean	Optional. If set to true, the Power Off menu item will be disabled when the user is logged in.

Key	Type	Value
<code>AdminHostInfo</code>	String	Optional. If this key is included in the payload, its value will be displayed as additional computer information on the login window. Before macOS 10.10, this string could contain only particular information (Host Name, System Version, and so on). After macOS 10.10, setting this key to any value will allow the user to click the "time" area of the menu bar to toggle through various computer information values.
<code>AllowList</code>	Array of Strings	Optional. User or group GUIDs of users that are allowed to log in. An asterisk "*" string specifies all users or groups.
<code>DenyList</code>	Array of Strings	Optional. User or group GUIDs of users that cannot log in. This list takes priority over the list in the AllowList.
<code>HideMobileAccounts</code>	Boolean	Optional. If set to true, mobile account users will not be visible in a user list. In some cases mobile users will show up as network users.
<code>ShutdownDisabled</code>	Boolean	Optional. If set to true, the Shut Down button item will be hidden.
<code>RestartDisabled</code>	Boolean	Optional. If set to true, the Restart item will be hidden.
<code>SleepDisabled</code>	Boolean	Optional. If set to true, the Sleep button item will be hidden.
<code>DisableConsoleAccess</code>	Boolean	Optional. If set to true, the Other user will disregard use of the '>console' special user name.
<code>LoginwindowText</code>	String	Optional. Text to display in the login window.
<code>ShutdownDisabledWhileLoggedIn</code>	Boolean	Optional. If set to true, the Shut Down menu item will be disabled when the user is logged in.
<code>RestartDisabledWhileLoggedIn</code>	Boolean	Optional. If set to true, the Restart menu item will be disabled when the user is logged in.
<code>PowerOffDisabledWhileLoggedIn</code>	Boolean	Optional. If set to true, the Power Off menu item will be disabled when the user is logged in.

Log Out Disabled While Logging In	Type	Value
LogOutDisabledWhileLoggingIn	Boolean	Optional. If set to true, this will disable the Log Out menu item when the user is logged in.
Availability		Available in macOS 10.13 and later.

Disable Screen Lock Immediately	Type	Value
DisableScreenLockImmediately	Boolean	Optional. If set to true, the immediate Screen Lock functions will be disabled.
Availability		Available in macOS 10.13 and later.

An older, separate, Loginwindow payload also exists and is designated by specifying `loginwindow payload` in the `key` field.

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

Key	Type	Value
Disable Login Items Subpayloads	Boolean	Optional. If set to true, the user is prevented from disabling login item launching using the Shift key.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

53

Log Out Disabled While Logging In	Type	Value
LogOutDisabledWhileLoggingIn	Boolean	Optional. If set to true, this will disable the Log Out menu item when the user is logged in.
Availability		Available in macOS 10.13 and later.

Disable Screen Lock Immediately	Type	Value
DisableScreenLockImmediately	Boolean	Optional. If set to true, the immediate Screen Lock functions will be disabled.
Availability		Available in macOS 10.13 and later.

An older, separate, Loginwindow payload also exists and is designated by specifying `loginwindow payload` in the `key` field.

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

Key	Type	Value
Disable Login Items Subpayloads	Boolean	Optional. If set to true, the user is prevented from disabling login item launching using the Shift key.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

53

Media Management

The profile configuration keys for media management are of two kinds: those that restrict disc burning and those that restrict media mounting and ejection. All keys are available on all versions of macOS and are supported on the user channel.

Disc Burning Payloads

Disc burning restrictions require both Disc Burning and Finder payloads.

The Disc Burning payload is designated by specifying `com.apple.DiscRestrictions`. In addition to the settings common to all payloads, this payload defines this key:

Key	Type	Value
BurnSupport	String	Required. Set to off to disable disc burning. Set to on for normal default operation. Set to authenticto require authentication. Setting this key to on will not enable disc burn support if it has already been disabled by other mechanisms or preferences.

The Finder payload is designated by specifying `com.apple.Burnable`.

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

Key	Type	Value
ProhibitBurn	Boolean	Required. Set to false to enable the Finder's burn support. Set to true to disable the Finder's burn support.

Allowed Media Payload

The Allowed Media payload is designated by specifying `com.apple.systemMountable`.

This payload defines these keys:

Key	Type	Value
logout-eject	Dictionary	Optional. Media type dictionary to define volumes to eject when the user logs out.
mount-control	Dictionary	Optional. Media type dictionary to control volume mounting.
unmount-control	Dictionary	Optional. Media type dictionary to control volume unmounting.

The Media type dictionaries can contain the following keys. Not all dictionaries use all keys. Values for media action strings are given in the next table.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

54

Media Management

The profile configuration keys for media management are of two kinds: those that restrict disc burning and those that restrict media mounting and ejection. All keys are available on all versions of macOS and are supported on the user channel.

Disc Burning Payloads

Disc burning restrictions require both Disc Burning and Finder payloads.

The Disc Burning payload is designated by specifying `com.apple.DiscRestrictions`. In addition to the settings common to all payloads, this payload defines this key:

Key	Type	Value
BurnSupport	String	Required. Set to off to disable disc burning. Set to on for normal default operation. Set to authenticto require authentication. Setting this key to on will not enable disc burn support if it has already been disabled by other mechanisms or preferences.

The Finder payload is designated by specifying `com.apple.Burnable`.

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

Key	Type	Value
ProhibitBurn	Boolean	Required. Set to false to enable the Finder's burn support. Set to true to disable the Finder's burn support.

Allowed Media Payload

The Allowed Media payload is designated by specifying `com.apple.systemMountable`.

This payload defines these keys:

Key	Type	Value
logout-eject	Dictionary	Optional. Media type dictionary to define volumes to eject when the user logs out.
mount-control	Dictionary	Optional. Media type dictionary to control volume mounting.
unmount-control	Dictionary	Optional. Media type dictionary to control volume unmounting.

The Media type dictionaries can contain the following keys. Not all dictionaries use all keys. Values for media action strings are given in the next table.

Key (media type)	Type	Value
all-media	String	Optional. Unused; set to empty string.
cd	String or Array of Strings	Optional. Media action string(s).
dvd	String or Array of Strings	Optional. Media action string(s).
bd	String or Array of Strings	Optional. Media action string(s).
blankcd	String or Array of Strings	Optional. Media action string(s).
blankdvd	String or Array of Strings	Optional. Media action string(s).
blankbd	String or Array of Strings	Optional. Media action string(s).
dvdram	String or Array of Strings	Optional. Media action string(s).
disk-image	String or Array of Strings	Optional. Media action string(s).
harddisk-internal	String or Array of Strings	Optional. Media action string(s).
networkdisk	String or Array of Strings	Optional. Media action string(s).
harddisk-external	String or Array of Strings	Optional. Media action string(s). Internally installed SD-Cards and USB flash drives are included in the harddisk-external category. This key is the default for media types that don't fall into other categories.

Media action strings are described below. You can combine some strings in arrays to create custom actions.

Key	Type	Value
authentic	Boolean	Optional. The user will be authenticated before the media is mounted.

Key (media type)	Type	Value
all-media	String	Optional. Unused; set to empty string.
cd	String or Array of Strings	Optional. Media action string(s).
dvd	String or Array of Strings	Optional. Media action string(s).
bd	String or Array of Strings	Optional. Media action string(s).
blankcd	String or Array of Strings	Optional. Media action string(s).
blankdvd	String or Array of Strings	Optional. Media action string(s).
blankbd	String or Array of Strings	Optional. Media action string(s).
dvdram	String or Array of Strings	Optional. Media action string(s).
disk-image	String or Array of Strings	Optional. Media action string(s).
harddisk-internal	String or Array of Strings	Optional. Media action string(s).
networkdisk	String or Array of Strings	Optional. Media action string(s).
harddisk-external	String or Array of Strings	Optional. Media action string(s). Internally installed SD-Cards and USB flash drives are included in the harddisk-external category. This key is the default for media types that don't fall into other categories.

Media action strings are described below. You can combine some strings in arrays to create custom actions.

Key	Type	Value
authentic	Boolean	Optional. The user will be authenticated before the media is mounted.

read-only	Boolean	Optional. The media will be mounted as read-only; this action cannot be combined with unmount controls.
deny	Boolean	Optional. The media will not be mounted.
eject	Boolean	Optional. The media will not be mounted and it will be ejected if possible. Note that some volumes are not defined as ejectable, so using the deny key may be the best solution. This action cannot be combined with unmount controls.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

55

read-only	Boolean	Optional. The media will be mounted as read-only; this action cannot be combined with unmount controls.
deny	Boolean	Optional. The media will not be mounted.
eject	Boolean	Optional. The media will not be mounted and it will be ejected if possible. Note that some volumes are not defined as ejectable, so using the deny key may be the best solution. This action cannot be combined with unmount controls.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

55

Mobile Accounts Payload

The Mobile Accounts payload is designated by specifying `com.apple.mobileaccountsPayload` value. This payload allows controls the authentication UI during mobile account creation. It must be installed as a device profile.

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

Key	Type	Value
<code>cachedaccounts</code>	Boolean	If set to <code>true</code> , the authentication UI is not displayed

`askForSecureTokenAuthByPass` when a mobile account is created. Suppressing the UI may prevent mobile accounts from being able to unlock a FileVault volume. Default is `false`

Availability: Available in macOS 10.13.5 and later.

Network Usage Rules Payload

The Network Usage Rules payload is designated by specifying `com.apple.networkusagerulesPayload` value.

Network Usage Rules allow enterprises to specify how managed apps use networks, such as cellular data networks. These rules only apply to managed apps.

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

Key	Type	Value
<code>ApplicationRules</code>	Array of Dictionaries	Required.

Each entry in the `ApplicationRules` field must be a dictionary containing these keys:

Key	Type	Value
<code>AppIdentifierArray</code>	Array	Optional. A list of managed app identifiers, as strings, that must follow the associated rules. If this key is missing, the rules will apply to all managed apps on the device.
		Each string in the <code>AppIdentifierArray</code> may either be an exact app identifier match, e.g. <code>com.mycompany.myapp</code> or it may specify a prefix match for the Bundle ID by using the * wildcard character. The wildcard character, if used, must appear after a period character (.), and may only appear once, at the end of the string, e.g. <code>com.mycompany.*</code>
<code>AllowRoamingCellData</code>	Boolean	Optional. Default <code>true</code> . If set to <code>false</code> matching managed apps will not be allowed to use cellular data when roaming.
<code>AllowCellularData</code>	Boolean	Optional. Default <code>true</code> . If set to <code>false</code> matching managed apps will not be allowed to use cellular data at any time.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

56

Mobile Accounts Payload

The Mobile Accounts payload is designated by specifying `com.apple.mobileaccountsPayload` value. This payload allows controls the authentication UI during mobile account creation. It must be installed as a device profile.

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

Key	Type	Value
<code>cachedaccounts</code>	Boolean	If set to <code>true</code> , the authentication UI is not displayed

`askForSecureTokenAuthByPass` when a mobile account is created. Suppressing the UI may prevent mobile accounts from being able to unlock a FileVault volume. Default is `false`

Availability: Available in macOS 10.13.5 and later.

Network Usage Rules Payload

The Network Usage Rules payload is designated by specifying `com.apple.networkusagerulesPayload` value.

Network Usage Rules allow enterprises to specify how managed apps use networks, such as cellular data networks. These rules only apply to managed apps.

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

Key	Type	Value
<code>ApplicationRules</code>	Array of Dictionaries	Required.

Each entry in the `ApplicationRules` field must be a dictionary containing these keys:

Key	Type	Value
<code>AppIdentifierArray</code>	Array	Optional. A list of managed app identifiers, as strings, that must follow the associated rules. If this key is missing, the rules will apply to all managed apps on the device.
		Each string in the <code>AppIdentifierArray</code> may either be an exact app identifier match, e.g. <code>com.mycompany.myapp</code> or it may specify a prefix match for the Bundle ID by using the * wildcard character. The wildcard character, if used, must appear after a period character (.), and may only appear once, at the end of the string, e.g. <code>com.mycompany.*</code>
<code>AllowRoamingCellData</code>	Boolean	Optional. Default <code>true</code> . If set to <code>false</code> matching managed apps will not be allowed to use cellular data when roaming.
<code>AllowCellularData</code>	Boolean	Optional. Default <code>true</code> . If set to <code>false</code> matching managed apps will not be allowed to use cellular data at any time.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

56

Notifications Payload

The Notifications Payload is designated by specifying `com.apple.notificationsPayload` value. It can contain only one payload, which must be installed on supervised devices. It is supported on the User Channel.

This payload specifies the restriction enforced notification settings for apps, using their bundle identifiers. It is supported on iOS 9.3 and later. In addition to the settings common to all payloads, this payload defines the following key:

Key	Type	Value
<code>NotificationSettings</code>	Array	Required. An array of dictionaries, each of which specifies notification settings for one bundle identifier.

Each entry in the `NotificationSettings` field contains the following dictionary:

Key	Type	Value
<code>BundleIdentifier</code>	String	Required. Bundle identifier of app to which to apply these notification settings.
<code>NotificationsEnabled</code>	Boolean	Optional. Whether notifications are allowed for this app. Default is <code>true</code> .

`ShowInNotificationCenter` Boolean

Notifications Payload

The Notifications Payload is designated by specifying `com.apple.notificationsPayload` value. It can contain only one payload, which must be installed on supervised devices. It is supported on the User Channel.

This payload specifies the restriction enforced notification settings for apps, using their bundle identifiers. It is supported on iOS 9.3 and later. In addition to the settings common to all payloads, this payload defines the following key:

Key	Type	Value
<code>NotificationSettings</code>	Array	Required. An array of dictionaries, each of which specifies notification settings for one bundle identifier.

Each entry in the `NotificationSettings` field contains the following dictionary:

Key	Type	Value
<code>BundleIdentifier</code>	String	Required. Bundle identifier of app to which to apply these notification settings.
<code>NotificationsEnabled</code>	Boolean	Optional. Whether notifications are allowed for this app. Default is <code>true</code> .

`ShowInNotificationCenter` Boolean

ShowInLockScreen	Boolean	Optional. Whether notifications can be shown in the lock screen. Default is true.
AlertType	Integer	Optional. The type of alert for notifications for this app: <ul style="list-style-type: none"> 0: None 1: Banner (default) 2: Modal Alert
BadgesEnabled	Boolean	Optional. Whether badges are allowed for this app. Default is true.
SoundsEnabled	Boolean	Optional. Whether sounds are allowed for this app. Default is true.
ShowInCarPlay	Boolean	Optional. Whether notifications can be shown in CarPlay. Default is true. Availability: Available in iOS 12 and later.
GroupingType	Integer	Optional. The type of grouping for notifications for this app: <ul style="list-style-type: none"> 0: Automatic - group notifications into app-specified groups. (Default) 1: By app - group notifications into one group. 2: Off - do not group notifications. Availability: Available in iOS 12 and later.
CriticalAlertEnabled	Boolean	Optional. Whether an app can mark a notification as a critical notification that will ignore Do Not Disturb and ringer settings. Default is false. Availability: Available in iOS 12 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

57

ShowInLockScreen	Boolean	Optional. Whether notifications can be shown in the lock screen. Default is true.
AlertType	Integer	Optional. The type of alert for notifications for this app: <ul style="list-style-type: none"> 0: None 1: Banner (default) 2: Modal Alert
BadgesEnabled	Boolean	Optional. Whether badges are allowed for this app. Default is true.
SoundsEnabled	Boolean	Optional. Whether sounds are allowed for this app. Default is true.
ShowInCarPlay	Boolean	Optional. Whether notifications can be shown in CarPlay. Default is true. Availability: Available in iOS 12 and later.
GroupingType	Integer	Optional. The type of grouping for notifications for this app: <ul style="list-style-type: none"> 0: Automatic - group notifications into app-specified groups. (Default) 1: By app - group notifications into one group. 2: Off - do not group notifications. Availability: Available in iOS 12 and later.
CriticalAlertEnabled	Boolean	Optional. Whether an app can mark a notification as a critical notification that will ignore Do Not Disturb and ringer settings. Default is false. Availability: Available in iOS 12 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

57

NSExtension Management

The NSExtension payload is designated by specifying `com.apple.NSExtensionPayload.adType`. This payload specifies which NSExtensions are allowed or disallowed on a system. Extensions can be managed by bundleID in whitelists and blacklists or by a blacklist of extension points.

It is supported on macOS 10.13 and later.

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

Key	Type	Value
AllowedExtensions	Optional. Array of extension identifiers for extensions that are allowed to run on the system.	
DeniedExtensions	Optional. Array of extension identifiers for extensions that are not allowed to run on the system.	
DeniedExtensionPoints	Optional. Array of NSExtension extension points for extensions that are not allowed to run on the system.	

If an array element within `DeniedExtensions` or `DeniedExtensionPoints`, `DeniedExtensionsWithPoints` be filled with a list of extension points that the client considers to be "public". These are the extension points referenced in developer documentation and supported by the Xcode programming environment.

Expansion of "AllPublicExtensionPoints" happens at evaluation time. The list of extension points may change from release to release.

This feature is intended as a way to specify "Start with no extensions belonging to any public extension points enabled but then allow only extensions A, B, C to run". Specifying "AllPublicExtensionPoints" will disallow both Apple and third-party extensions within the "public" extension points but will still allow extensions belonging to system-critical extension points to execute.

Parental Controls Payload

Parental Control on macOS consists of many different payloads which are set individually depending on the type of control required. Parental control payloads are supported on the user channel. Each payload and its respective keys are described in the sections below.

Parental Control Web Content Filter Payload

The Parental Control Web Content Filter payload is designated by specifying `com.apple.familycontrols` as the `Payload` value.

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

Key	Type	Value
restrictWeb	Boolean	Required. Set to true to enable Web content filters.
useContentFilter	Boolean	Optional. Set to true to try to automatically filter content.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

58

NSExtension Management

The NSExtension payload is designated by specifying `com.apple.NSExtensionPayload.adType`. This payload specifies which NSExtensions are allowed or disallowed on a system. Extensions can be managed by bundleID in whitelists and blacklists or by a blacklist of extension points.

It is supported on macOS 10.13 and later.

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

Key	Type	Value
AllowedExtensions	Optional. Array of extension identifiers for extensions that are allowed to run on the system.	
DeniedExtensions	Optional. Array of extension identifiers for extensions that are not allowed to run on the system.	
DeniedExtensionPoints	Optional. Array of NSExtension extension points for extensions that are not allowed to run on the system.	

If an array element within `DeniedExtensions` or `DeniedExtensionPoints`, `DeniedExtensionsWithPoints` be filled with a list of extension points that the client considers to be "public". These are the extension points referenced in developer documentation and supported by the Xcode programming environment.

Expansion of "AllPublicExtensionPoints" happens at evaluation time. The list of extension points may change from release to release.

This feature is intended as a way to specify "Start with no extensions belonging to any public extension points enabled but then allow only extensions A, B, C to run". Specifying "AllPublicExtensionPoints" will disallow both Apple and third-party extensions within the "public" extension points but will still allow extensions belonging to system-critical extension points to execute.

Parental Controls Payload

Parental Control on macOS consists of many different payloads which are set individually depending on the type of control required. Parental control payloads are supported on the user channel. Each payload and its respective keys are described in the sections below.

Parental Control Web Content Filter Payload

The Parental Control Web Content Filter payload is designated by specifying `com.apple.familycontrols` as the `Payload` value.

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

Key	Type	Value
restrictWeb	Boolean	Required. Set to true to enable Web content filters.
useContentFilter	Boolean	Optional. Set to true to try to automatically filter content.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

58

Key	Type	Value
whiteListEnabled	Boolean	Optional. Set to true to use the filterWhiteList and filterBlackList lists.
siteWhitelist	Dictionary	Required if <code>whiteListEnabled</code> is true. This key contains an array of dictionaries (see below) that define additional allowed sites besides those in the automated list of allowed and unallowed sites, including disallowed adult sites.
filterWhiteList	Array of URL Strings	Optional. If specified and <code>restrictWeb</code> is true, an array of URLs designating the only allowed Websites.
filterBlackList	Array of URL Strings	Optional. If specified and <code>restrictWeb</code> is true, an array of URLs of Websites never to be allowed.

Each `siteWhitelist` contains these keys:

Key	Type	Value
address	String	Required. Site prefix, including <code>http</code> (scheme).
pageTitle	String	Optional. Site page title.

Parental Control Time Limits Payload

Key	Type	Value
whiteListEnabled	Boolean	Optional. Set to true to use the filterWhiteList and filterBlackList lists.
siteWhitelist	Dictionary	Required if <code>whiteListEnabled</code> is true. This key contains an array of dictionaries (see below) that define additional allowed sites besides those in the automated list of allowed and unallowed sites, including disallowed adult sites.
filterWhiteList	Array of URL Strings	Optional. If specified and <code>restrictWeb</code> is true, an array of URLs designating the only allowed Websites.
filterBlackList	Array of URL Strings	Optional. If specified and <code>restrictWeb</code> is true, an array of URLs of Websites never to be allowed.

Each `siteWhitelist` contains these keys:

Key	Type	Value
address	String	Required. Site prefix, including <code>http</code> (scheme).
pageTitle	String	Optional. Site page title.

The Parental Control Time Limits payload is designated by specifying `com.apple.familycontrols.timeLimits` as the `Payload` `value`.

It consists of a dictionary containing a master enabled flag plus a dictionary of time limit specification keys.

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

Key	Type	Value
<code>familyControls</code>	Boolean	Required. Set to <code>true</code> to use time limits.

`timeLimits` Dictionary Required if `familyControls` is `true`. The `timeLimits` settings.

Each `timeLimits` dictionary contains these keys:

Key	Type	Value
<code>weekdayAllowance</code>	Dictionary	Optional. Weekday allowance settings.
<code>weekdayCurfew</code>	Dictionary	Optional. Weekday curfew settings.
<code>weekendAllowance</code>	Dictionary	Optional. Weekend allowance settings.
<code>weekendCurfew</code>	Dictionary	Optional. Weekend curfew settings.

Each `allowOrRefuse` dictionary contains these keys:

Key	Type	Value
<code>enabled</code>	Boolean	Required. Set to <code>true</code> to enable these settings.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

59

The Parental Control Time Limits payload is designated by specifying `com.apple.familycontrols.timeLimits` as the `Payload` `value`.

It consists of a dictionary containing a master enabled flag plus a dictionary of time limit specification keys.

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

Key	Type	Value
<code>familyControls</code>	Boolean	Required. Set to <code>true</code> to use time limits.

`timeLimits` Dictionary Required if `familyControls` is `true`. The `timeLimits` settings.

Each `timeLimits` dictionary contains these keys:

Key	Type	Value
<code>weekdayAllowance</code>	Dictionary	Optional. Weekday allowance settings.
<code>weekdayCurfew</code>	Dictionary	Optional. Weekday curfew settings.
<code>weekendAllowance</code>	Dictionary	Optional. Weekend allowance settings.
<code>weekendCurfew</code>	Dictionary	Optional. Weekend curfew settings.

Each `allowOrRefuse` dictionary contains these keys:

Key	Type	Value
<code>enabled</code>	Boolean	Required. Set to <code>true</code> to enable these settings.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

59

Key	Type	Value
<code>rangeType</code>	Integer	Required. Type of day range: 0 = weekday, 1 = weekend.
<code>start</code>	String	Optional. Curfew start time in the format <code>%d:%d:%d</code> .
<code>end</code>	String	Optional. Curfew end time in the format <code>%d:%d:%d</code> .
<code>secondsPerDay</code>	Integer	Optional. Seconds for that day for allowance.

Parental Control Application Access Payload

The Parental Control Application Access payload is designated by specifying `com.apple.applicationAccess` as the `Payload` `value`.

It enables application access restrictions on macOS.

To determine if an application can be launched, these rules are evaluated:

1. Certain system applications and utilities are always allowed to run.
2. The `whiteList` is searched to see if a matching entry is found by `bundleID`. If a match is found, the `appID` and `detachedSignature` (if present) are used to verify the signature of the application being launched. If the signature is valid and matches the designated requirement (in the `appKey`), the application is allowed to launch.
3. If the path to the binary being launched matches (or is in a subdirectory) of a path in `pathBlackList` the binary is denied.
4. If the path to the binary being launched matches (or is a subdirectory) of a path in `pathWhiteList` the binary is allowed to launch.
5. The binary is denied permission to launch.

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

Key	Type	Value
<code>familyControlsEnabled</code>	Boolean	Required. Set to <code>true</code> to enable application access restrictions.
<code>whiteList</code>	Array of Dictionaries	Optional. A list of code signatures for applications that are allowed to run.
<code>pathBlackList</code>	Array of Strings	Optional. Paths to disallowed applications.
<code>pathWhiteList</code>	Array of Strings	Optional. Paths to allowed applications.

Each `whiteList` dictionary contains these keys:

Key	Type	Value
<code>bundleID</code>	String	Required. Bundle ID of application.
<code>appID</code>	Data	Required. The designated requirement describing the code signature of this executable. This value is obtained from the <code>Security.framework</code> using <code>SecurityCopyDesignatedRequirement</code> .

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

60

Each `whiteList` dictionary contains these keys:

Key	Type	Value
<code>familyControlsEnabled</code>	Boolean	Required. Set to <code>true</code> to enable application access restrictions.
<code>whiteList</code>	Array of Dictionaries	Optional. A list of code signatures for applications that are allowed to run.
<code>pathBlackList</code>	Array of Strings	Optional. Paths to disallowed applications.
<code>pathWhiteList</code>	Array of Strings	Optional. Paths to allowed applications.

Each `whiteList` dictionary contains these keys:

Key	Type	Value
<code>bundleID</code>	String	Required. Bundle ID of application.
<code>appID</code>	Data	Required. The designated requirement describing the code signature of this executable. This value is obtained from the <code>Security.framework</code> using <code>SecurityCopyDesignatedRequirement</code> .

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

60

Key	Type	Value
<code>detachedSignature</code>	Optional.	Can be used to provide the required signature for an unsigned binary. Generate an ad-hoc signature of the unsigned binary and store the signature here.
<code>disabled</code>	Boolean	Optional. Specifies whether this application information is to be included in the <code>whiteList</code> . Set to <code>true</code> to keep the application off the <code>whiteList</code> . It could still be allowed to launch via <code>pathWhiteList</code> although this behavior is discouraged. Default is <code>false</code> .
<code>subApps</code>	Array of Dictionaries	Optional. For applications that include nested helper applications, describes the signatures of embedded applications. The dictionary format is the same as for the <code>whiteList</code> .
<code>displayName</code>	String	Optional. Display name.

Parental Control Dashboard Payload

The Parental Control Dashboard payload is designated by specifying `com.apple.dashboardBoardType`

Key	Type	Value
<code>detachedSignature</code>	Optional.	Can be used to provide the required signature for an unsigned binary. Generate an ad-hoc signature of the unsigned binary and store the signature here.
<code>disabled</code>	Boolean	Optional. Specifies whether this application information is to be included in the <code>whiteList</code> . Set to <code>true</code> to keep the application off the <code>whiteList</code> . It could still be allowed to launch via <code>pathWhiteList</code> although this behavior is discouraged. Default is <code>false</code> .
<code>subApps</code>	Array of Dictionaries	Optional. For applications that include nested helper applications, describes the signatures of embedded applications. The dictionary format is the same as for the <code>whiteList</code> .
<code>displayName</code>	String	Optional. Display name.

Parental Control Dashboard Payload

The Parental Control Dashboard payload is designated by specifying `com.apple.dashboardBoardType`

value.

It is used to define a white list of dashboard widgets.

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

Key	Type	Value
whiteListEnabled	Required. Set to true to enable the widget white list items.	
whiteList	Array of Dictionaries	Required. List that defines Dashboard widgets.

Each widget whiteList dictionary contains these keys:

Key	Type	Value
Type	String	Required. Set to bundle to use a widget's bundle ID as its ID.
ID	String	Required. The bundle ID of a widget.

Parental Control Dictionary Payload

The Parental Control Dictionary payload is designated by specifying com.apple.DictionaryPayload as the value.

It enables the restrictions defined in the device's Parental Controls Dictionary.

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

Key	Type	Value
parentalControlEnabled	Required. Set to true to enable parental controls dictionary restrictions.	

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

61

value.

It is used to define a white list of dashboard widgets.

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

Key	Type	Value
whiteListEnabled	Required. Set to true to enable the widget white list items.	
whiteList	Array of Dictionaries	Required. List that defines Dashboard widgets.

Each widget whiteList dictionary contains these keys:

Key	Type	Value
Type	String	Required. Set to bundle to use a widget's bundle ID as its ID.
ID	String	Required. The bundle ID of a widget.

Parental Control Dictionary Payload

The Parental Control Dictionary payload is designated by specifying com.apple.DictionaryPayload as the value.

It enables the restrictions defined in the device's Parental Controls Dictionary.

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

Key	Type	Value
parentalControlEnabled	Required. Set to true to enable parental controls dictionary restrictions.	

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

61

Parental Control Dictation and Profanity Payload

The Parental Control Dictation and Profanity payload is designated by specifying com.apple.ironwood.support as the Payload value.

It disables dictation and suppresses profanity on the device.

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

Key	Type	Value
IronwoodAllowed	Boolean	Optional. Set to false to disable dictation.
ProfanityAllowed	Boolean	Optional. Set to false to suppress profanity.

Parental Control Game Center Payload

The Parental Control Game Center payload is designated by specifying com.apple.aggregatorLoadType as the value.

It restricts Game Center options on the device.

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

Key	Type	Value
GKFeatureGameCenterAllowed	Boolean	Optional. Set to false to disable Game Center.
GKFeatureAccountModificationsAllowed	Boolean	Optional. Set to false to disable account modifications.
GKFeatureAddingGameCenterFriendsAllowed	Boolean	Optional. Set to false to disable adding Game Center friends.
GKFeatureMultiplayerGamingAllowed	Boolean	Optional. Set to false to disable multiplayer gaming.

Additional Parental Controls

Additional parental control functions can be found in the following payloads:

- System Policy Control Payload
- Email Payload
- Media Management
- AppStore Payload
- Dock Payload

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

62

Parental Control Dictation and Profanity Payload

The Parental Control Dictation and Profanity payload is designated by specifying com.apple.ironwood.support as the Payload value.

It disables dictation and suppresses profanity on the device.

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

Key	Type	Value
IronwoodAllowed	Boolean	Optional. Set to false to disable dictation.
ProfanityAllowed	Boolean	Optional. Set to false to suppress profanity.

Parental Control Game Center Payload

The Parental Control Game Center payload is designated by specifying com.apple.aggregatorLoadType as the value.

It restricts Game Center options on the device.

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

Key	Type	Value
GKFeatureGameCenterAllowed	Boolean	Optional. Set to false to disable Game Center.
GKFeatureAccountModificationsAllowed	Boolean	Optional. Set to false to disable account modifications.
GKFeatureAddingGameCenterFriendsAllowed	Boolean	Optional. Set to false to disable adding Game Center friends.
GKFeatureMultiplayerGamingAllowed	Boolean	Optional. Set to false to disable multiplayer gaming.

Additional Parental Controls

Additional parental control functions can be found in the following payloads:

- System Policy Control Payload
- Email Payload
- Media Management
- AppStore Payload
- Dock Payload

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

62

Passcode Policy Payload

The Passcode Policy payload is designated by specifying com.apple.mobiledevice.passcodePolicy as the value.

The presence of this payload type prompts an iOS or macOS device to present the user with a passcode entry mechanism. The complexity of the passcode can be customized with this payload.

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".

Passcode Policy Payload

The Passcode Policy payload is designated by specifying com.apple.mobiledevice.passcodePolicy as the value.

The presence of this payload type prompts an iOS or macOS device to present the user with a passcode entry mechanism. The complexity of the passcode can be customized with this payload.

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	Optional. Default 11. Allowed range [2..11]. Specifies the number of allowed failed attempts to enter the passcode at the device's lock screen. After six failed attempts, there is a time delay imposed before a passcode can be entered again. The delay increases with each attempt. On macOS, set <code>minutesUntilFailure</code> and <code>delays</code> to before the next passcode can be entered. Once this number is exceeded, on macOS the device is locked and on iOS the device is wiped.	
minutesUntilLoginReset	Optional. On macOS, this can be set to the number of minutes before the login will be reset after the <code>maxFailedAttempts</code> has been reached. This key requires setting <code>maxFailedAttempts</code> . Availability: Available in macOS 10.10 and later.	
maxInactivity	Optional. Default Infinity. Specifies the maximum 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. The user can edit this setting, but the value cannot exceed the <code>maxInactivity</code> . In macOS, this will be translated to screensaver settings.	
maxPINAgeInDays	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	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 %\$#.	
minLength	Integer	Optional. Default 0. Specifies the minimum overall length of the passcode. This parameter is independent of the also optional <code>minComplexChars</code> argument.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

63

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	Optional. Default 11. Allowed range [2..11]. Specifies the number of allowed failed attempts to enter the passcode at the device's lock screen. After six failed attempts, there is a time delay imposed before a passcode can be entered again. The delay increases with each attempt. On macOS, set <code>minutesUntilFailure</code> and <code>delays</code> to before the next passcode can be entered. Once this number is exceeded, on macOS the device is locked and on iOS the device is wiped.	
minutesUntilLoginReset	Optional. On macOS, this can be set to the number of minutes before the login will be reset after the <code>maxFailedAttempts</code> has been reached. This key requires setting <code>maxFailedAttempts</code> . Availability: Available in macOS 10.10 and later.	
maxInactivity	Optional. Default Infinity. Specifies the maximum 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. The user can edit this setting, but the value cannot exceed the <code>maxInactivity</code> . In macOS, this will be translated to screensaver settings.	
maxPINAgeInDays	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	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 %\$#.	
minLength	Integer	Optional. Default 0. Specifies the minimum overall length of the passcode. This parameter is independent of the also optional <code>minComplexChars</code> argument.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

63

Key	Type	Value
requireAlphanumeric	Boolean	Optional. Specifies whether the user must also enter alphabetic characters ("abcd") along with numbers, or if numbers only are sufficient. Default is <code>false</code>
pinHistory	Integer	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.
maxGracePeriod	Integer	Optional. The maximum grace period, in minutes, to unlock without entering a passcode. Default is 0, that is no grace period, which requires entering a passcode immediately. In macOS, this will be translated to screensaver settings.
allowFingerPrintModification	Boolean	Optional. Supervised only. Not supported on macOS. Allows the user to modify Touch ID. Default is <code>false</code>
changeAtNextBoot	Boolean	Optional. On macOS, setting this to <code>true</code> will cause a password reset to occur the next time the user tries to authenticate. If this key is set in a device profile, the setting takes effect for all users, and admin authentications may fail until the admin user password is also reset. Availability: Available in macOS 10.13 and later.

Privacy Preferences Policy Control Payload

The Privacy Preferences payload is designated by specifying `com.apple.TCC.configuration-profile` as the `Payload Value`

It controls the settings that are displayed in the "Privacy" tab of the "Security & Privacy" pane in System Preferences.

This profile must be delivered via a user approved MDM server in a device profile.

Availability: Available in macOS 10.14 and later.

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

Key	Type	Value
ServiceDictionary	Dictionary	Keys are limited to the privacy service names listed below. Each key is an array of dictionaries describing the app or process to which access is given. In the case of conflicting specifications, the most restrictive setting (deny) will be used.

Privacy Service Dictionary Keys

Key	Type	Value
AddressBook	Array of Identity Dictionaries	Optional. Contact information managed by Contacts.app.
Calendar	Array of Identity Dictionaries	Optional. Calendar information managed by Calendar.app.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

64

Key	Type	Value
requireAlphanumeric	Boolean	Optional. Specifies whether the user must also enter alphabetic characters ("abcd") along with numbers, or if numbers only are sufficient. Default is <code>false</code>
pinHistory	Integer	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.
maxGracePeriod	Integer	Optional. The maximum grace period, in minutes, to unlock without entering a passcode. Default is 0, that is no grace period, which requires entering a passcode immediately. In macOS, this will be translated to screensaver settings.
allowFingerPrintModification	Boolean	Optional. Supervised only. Not supported on macOS. Allows the user to modify Touch ID. Default is <code>false</code>
changeAtNextBoot	Boolean	Optional. On macOS, setting this to <code>true</code> will cause a password reset to occur the next time the user tries to authenticate. If this key is set in a device profile, the setting takes effect for all users, and admin authentications may fail until the admin user password is also reset. Availability: Available in macOS 10.13 and later.

Privacy Preferences Policy Control Payload

The Privacy Preferences payload is designated by specifying `com.apple.TCC.configuration-profile` as the `Payload Value`

It controls the settings that are displayed in the "Privacy" tab of the "Security & Privacy" pane in System Preferences.

This profile must be delivered via a user approved MDM server in a device profile.

Availability: Available in macOS 10.14 and later.

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

Key	Type	Value
ServiceDictionary	Dictionary	Keys are limited to the privacy service names listed below. Each key is an array of dictionaries describing the app or process to which access is given. In the case of conflicting specifications, the most restrictive setting (deny) will be used.

Privacy Service Dictionary Keys

Key	Type	Value
AddressBook	Array of Identity Dictionaries	Optional. Contact information managed by Contacts.app.
Calendar	Array of Identity Dictionaries	Optional. Calendar information managed by Calendar.app.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

64

Key	Type	Value
Reminders	Array of Identity Dictionaries	Optional. Reminders information managed by Reminders.app.
Photos	Array of Identity Dictionaries	Optional. Pictures managed by Photos.app in <code>~/Pictures/.photoslibrary</code>
Camera	Array of Identity Dictionaries	Optional. A system camera. Access to the camera cannot be given in a profile; it can only be denied.
Microphone	Array of Identity Dictionaries	Optional. A system microphone. Access to the microphone cannot be given in a profile; it can only be denied.

Key	Type	Value
Reminders	Array of Identity Dictionaries	Optional. Reminders information managed by Reminders.app.
Photos	Array of Identity Dictionaries	Optional. Pictures managed by Photos.app in <code>~/Pictures/.photoslibrary</code>
Camera	Array of Identity Dictionaries	Optional. A system camera. Access to the camera cannot be given in a profile; it can only be denied.
Microphone	Array of Identity Dictionaries	Optional. A system microphone. Access to the microphone cannot be given in a profile; it can only be denied.

Accessibility	Array of Identity Dictionaries	Optional. Control the application via the Accessibility subsystem.
PostEvent	Array of Identity Dictionaries	Optional. Allows the application to use CoreGraphics APIs to send CGEvents to the system event stream.
SystemPolicyAllowFile	Array of Identity Dictionaries	Optional. Allows the application access to all protected files, including system administration files.
SystemPolicySystemAdminFile	Array of Identity Dictionaries	Optional. Allows the application access to some files used in system administration.
AppleEvents	Array of Identity Dictionaries	Optional. Allows the application to send a restricted AppleEvent to another process.

Identity Dictionary Keys

Key	Type	Value
Identifier	String	The bundle ID or installation path of the binary.
IdentifierType	String	The type of Identifier value. Must be either bundle or ID path. Application bundles should be identified by bundle ID. Non-bundled binaries must be identified by installation path. Helper tools embedded within an application bundle will automatically inherit the permissions of their enclosing app bundle.
CodeRequirement	String	Obtained via the command "code design -display -r -".
StaticCode	Boolean	Optional. If set to true, statically validate the code requirement. Used only if the process invalidates its dynamic code signature. Defaults to false.
Allowed	Boolean	If set to true, access is granted. Otherwise the process does not have access. The user is not prompted and cannot change this value.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

65

Accessibility	Array of Identity Dictionaries	Optional. Control the application via the Accessibility subsystem.
PostEvent	Array of Identity Dictionaries	Optional. Allows the application to use CoreGraphics APIs to send CGEvents to the system event stream.
SystemPolicyAllowFile	Array of Identity Dictionaries	Optional. Allows the application access to all protected files, including system administration files.
SystemPolicySystemAdminFile	Array of Identity Dictionaries	Optional. Allows the application access to some files used in system administration.
AppleEvents	Array of Identity Dictionaries	Optional. Allows the application to send a restricted AppleEvent to another process.

Identity Dictionary Keys

Key	Type	Value
Identifier	String	The bundle ID or installation path of the binary.
IdentifierType	String	The type of Identifier value. Must be either bundle or ID path. Application bundles should be identified by bundle ID. Non-bundled binaries must be identified by installation path. Helper tools embedded within an application bundle will automatically inherit the permissions of their enclosing app bundle.
CodeRequirement	String	Obtained via the command "code design -display -r -".
StaticCode	Boolean	Optional. If set to true, statically validate the code requirement. Used only if the process invalidates its dynamic code signature. Defaults to false.
Allowed	Boolean	If set to true, access is granted. Otherwise the process does not have access. The user is not prompted and cannot change this value.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

65

Key	Type	Value
AEReceiverIdentifier	String	Optional. The identifier of the process receiving an AppleEvent by the Identifier. Required for AppleEvents not valid for other services.
AEReceiverIdentifierType	String	Optional. The type of AEReceiverIdentifier. Must be either bundle or path. Required for AppleEvents service; not valid for other services.
AEReceiverCodeRequirement	String	Optional. Code requirement for the receiving binary. Required for AppleEvents not valid for other services.
Comment	String	Not used.

Profile Removal Password Payload

The Removal Password payload is designated by specifying com.apple.profileRemovalPassword as the password. Pay load 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. Supervised only. Specifies the removal password for the profile.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

66

Key	Type	Value
AEReceiverIdentifier	String	Optional. The identifier of the process receiving an AppleEvent by the Identifier. Required for AppleEvents not valid for other services.
AEReceiverIdentifierType	String	Optional. The type of AEReceiverIdentifier. Must be either bundle or path. Required for AppleEvents service; not valid for other services.
AEReceiverCodeRequirement	String	Optional. Code requirement for the receiving binary. Required for AppleEvents not valid for other services.
Comment	String	Not used.

Profile Removal Password Payload

The Removal Password payload is designated by specifying com.apple.profileRemovalPassword as the password. Pay load 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. Supervised only. Specifies the removal password for the profile.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

66

Restrictions Payload

The Restrictions payload is designated by specifying com.apple.applicationRestrictions as the type value.

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

Note

Restrictions Payload

The Restrictions payload is designated by specifying com.apple.applicationRestrictions as the type value.

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

Note

You can specify additional restrictions, including maximum allowed content ratings, by creating a profile using Apple Configurator 2 or Profile Manager.

The Restrictions payload is supported in iOS; some keys are also supported in macOS, as noted below.

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

Key	Type	Value
allowAccountModification	Boolean	Optional. Supervised only. If set to <code>1</code> a <code>\$account</code> modification is disabled. Availability: Available only in iOS 7.0 and later.
allowAddingGameFriends	Boolean	Optional. When <code>f</code> a <code>1</code> prohibits adding friends to Game Center. This key is deprecated on unsupervised devices.
allowAirDrop	Boolean	Optional. Supervised only. If set to <code>1</code> a <code>\$AirDrop</code> is disabled. Availability: Available only in iOS 7.0 and later.
allowAppCellularModification	Boolean	Optional. Supervised only. If set to <code>1</code> a <code>\$changes</code> to cellular data usage for apps are disabled. Availability: Available only in iOS 7.0 and later.
allowAppInstall	Boolean	Optional. Supervised only. When <code>f</code> a <code>1</code> <code>\$the</code> App Store is disabled and its icon is removed from the Home screen. Users are unable to install or update their applications. This key is deprecated on unsupervised devices. In iOS 10 and later, MDM commands can override this restriction.
allowAppRemoval	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> removal of apps from iOS device. This key is deprecated on unsupervised devices.
allowAssistant	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> Siri. Defaults to <code>true</code>
allowAssistantUGeneratedContent	Boolean	Optional. Supervised only. When <code>f</code> a <code>1</code> prevents Siri from querying user-generated content from the web. Availability: Available in iOS 7 and later.
allowAssistantWithLocked	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$the</code> user is unable to use Siri when the device is locked. Defaults to <code>true</code> . 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 <code>1</code> a <code>\$Apple Books</code> will be disabled. This will default to <code>true</code> . Availability: Available in iOS 6.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

67

You can specify additional restrictions, including maximum allowed content ratings, by creating a profile using Apple Configurator 2 or Profile Manager.

The Restrictions payload is supported in iOS; some keys are also supported in macOS, as noted below.

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

Key	Type	Value
allowAccountModification	Boolean	Optional. Supervised only. If set to <code>1</code> a <code>\$account</code> modification is disabled. Availability: Available only in iOS 7.0 and later.
allowAddingGameFriends	Boolean	Optional. When <code>f</code> a <code>1</code> prohibits adding friends to Game Center. This key is deprecated on unsupervised devices.
allowAirDrop	Boolean	Optional. Supervised only. If set to <code>1</code> a <code>\$AirDrop</code> is disabled. Availability: Available only in iOS 7.0 and later.
allowAppCellularModification	Boolean	Optional. Supervised only. If set to <code>1</code> a <code>\$changes</code> to cellular data usage for apps are disabled. Availability: Available only in iOS 7.0 and later.
allowAppInstall	Boolean	Optional. Supervised only. When <code>f</code> a <code>1</code> <code>\$the</code> App Store is disabled and its icon is removed from the Home screen. Users are unable to install or update their applications. This key is deprecated on unsupervised devices. In iOS 10 and later, MDM commands can override this restriction.
allowAppRemoval	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> removal of apps from iOS device. This key is deprecated on unsupervised devices.
allowAssistant	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> Siri. Defaults to <code>true</code>
allowAssistantUGeneratedContent	Boolean	Optional. Supervised only. When <code>f</code> a <code>1</code> prevents Siri from querying user-generated content from the web. Availability: Available in iOS 7 and later.
allowAssistantWithLocked	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$the</code> user is unable to use Siri when the device is locked. Defaults to <code>true</code> . 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 <code>1</code> a <code>\$Apple Books</code> will be disabled. This will default to <code>true</code> . Availability: Available in iOS 6.0 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

67

Key	Type	Value
allowBookstoreErotica	Boolean	Optional. Supervised only prior to iOS 6.1. If set to <code>1</code> <code>\$the</code> user will not be able to download media from Apple Books that has been tagged as erotica. This will default to <code>true</code> . Availability: Available in iOS and in tvOS 11.3 and later.
allowCamera	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$the</code> camera is completely disabled and its icon is removed from the Home screen. Users are unable to take photographs. Availability: Available in iOS and in macOS 10.11 and later.
allowChat	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> the use of iMessage with supervised devices. If the device supports text messaging, the user can still send and receive text messages. Availability: Available in iOS 6.0 and later.
allowCloudBackup	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> backing up the device to iCloud. Availability: Available in iOS 5.0 and later.
allowCloudBookmark	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Bookmark sync. Availability: Available in macOS 10.12 and later.
allowCloudMail	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS Mail iCloud services. Availability: Available in macOS 10.12 and later.
allowCloudCalendar	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Calendar services. Availability: Available in macOS 10.12 and later.
allowCloudReminder	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> iCloud Reminder services. Availability: Available in macOS 10.12 and later.
allowCloudAddressBook	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Address Book services. Availability: Available in macOS 10.12 and later.
allowCloudNotes	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Notes services. Availability: Available in macOS 10.12 and later.
allowCloudDocumentSync	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> document and key-value syncing to iCloud. This key is deprecated on unsupervised devices. Availability: Available in iOS 5.0 and later and in macOS 10.11 and later.
allowCloudKeychainSync	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> iCloud keychain synchronization. Default is <code>true</code> Availability: Available in iOS 7.0 and later and macOS 10.12 and later.
allowContentCaching	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$this</code> disallows content caching. Defaults to <code>true</code> Availability: Available only in macOS 10.13 and later.
allowDiagnosticSubmission	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$this</code> prevents the device from automatically submitting diagnostic reports to Apple. Defaults to <code>true</code> Availability: Available only in iOS 6.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

68

Key	Type	Value
allowBookstoreErotica	Boolean	Optional. Supervised only prior to iOS 6.1. If set to <code>1</code> <code>\$the</code> user will not be able to download media from Apple Books that has been tagged as erotica. This will default to <code>true</code> . Availability: Available in iOS and in tvOS 11.3 and later.
allowCamera	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$the</code> camera is completely disabled and its icon is removed from the Home screen. Users are unable to take photographs. Availability: Available in iOS and in macOS 10.11 and later.
allowChat	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> the use of iMessage with supervised devices. If the device supports text messaging, the user can still send and receive text messages. Availability: Available in iOS 6.0 and later.
allowCloudBackup	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> backing up the device to iCloud. Availability: Available in iOS 5.0 and later.
allowCloudBookmark	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Bookmark sync. Availability: Available in macOS 10.12 and later.
allowCloudMail	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS Mail iCloud services. Availability: Available in macOS 10.12 and later.
allowCloudCalendar	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Calendar services. Availability: Available in macOS 10.12 and later.
allowCloudReminder	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> iCloud Reminder services. Availability: Available in macOS 10.12 and later.
allowCloudAddressBook	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Address Book services. Availability: Available in macOS 10.12 and later.
allowCloudNotes	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disallows</code> macOS iCloud Notes services. Availability: Available in macOS 10.12 and later.
allowCloudDocumentSync	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> document and key-value syncing to iCloud. This key is deprecated on unsupervised devices. Availability: Available in iOS 5.0 and later and in macOS 10.11 and later.
allowCloudKeychainSync	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$disables</code> iCloud keychain synchronization. Default is <code>true</code> Availability: Available in iOS 7.0 and later and macOS 10.12 and later.
allowContentCaching	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$this</code> disallows content caching. Defaults to <code>true</code> Availability: Available only in macOS 10.13 and later.
allowDiagnosticSubmission	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$this</code> prevents the device from automatically submitting diagnostic reports to Apple. Defaults to <code>true</code> Availability: Available only in iOS 6.0 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

68

Key	Type	Value
allowExplicitContent	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$explicit</code> 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. This key is deprecated on unsupervised devices.

Key	Type	Value
allowExplicitContent	Boolean	Optional. When <code>f</code> a <code>1</code> <code>\$explicit</code> 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. This key is deprecated on unsupervised devices.

<code>allowFindMyFriendsModification</code>	Availability: Available in iOS and in tvOS 11.3 and later. Optional. Supervised only. If set to <code>1</code> changes to Find My Friends are disabled.
<code>allowFingerprintUnlock</code>	Availability: Available only in iOS 7.0 and later. Optional. If <code>1</code> prevents Touch ID from unlocking a device.
<code>allowGameCenter</code>	Availability: Available in iOS 7 and later and in macOS 10.12.4 and later. Optional. Supervised only. When <code>1</code> Game Center is disabled and its icon is removed from the Home screen. Default is <code>true</code>
<code>allowGlobalBackgroundFetchWhenRoaming</code>	Availability: Available only in iOS 6.0 and later. Optional. When <code>1</code> disables global background fetch activity when an iOS phone is roaming.
<code>allowInAppPurchasing</code>	Availability: Available only in iOS 7 and later. Optional. When <code>1</code> prohibits in-app purchasing.
<code>allowLockScreen</code>	Availability: Available only in iOS 7 and later. Optional. If <code>1</code> prevents Control Center from appearing on the Lock screen.
<code>allowHostPairing</code>	Availability: Available only in iOS 7 and later. Supervised only. If set to <code>1</code> host pairing is disabled with the exception of the supervision host. If no supervision host certificate has been configured, all pairing is disabled. Host pairing lets the administrator control which devices an iOS 7 device can pair with.
<code>allowLockScreenNotificationsView</code>	Availability: Available only in iOS 7.0 and later. Optional. If set to <code>1</code> Notifications history view on the lock screen is disabled and users can't view past notifications. Though, when the device is locked, the user will still be able to view notifications when they arrive.
<code>allowLockScreenView</code>	Availability: Available only in iOS 7.0 and later. Optional. If set to <code>1</code> Today view in Notification Center on the lock screen is disabled.
<code>allowMultiplayer</code>	Availability: Available only in iOS 7.0 and later. Optional. When <code>1</code> prohibits multiplayer gaming. This key is deprecated on unsupervised devices.
<code>allowOpenFromManagedToUnmanaged</code>	Availability: Available only in iOS 7.0 and later. Optional. If <code>1</code> documents in managed apps and accounts only open in other managed apps and accounts. Default is <code>true</code>
<code>allowOpenFromUnmanagedToManaged</code>	Availability: Available only in iOS 7.0 and later. Optional. If set to <code>1</code> documents in unmanaged apps and accounts will only open in other unmanaged apps and accounts. Default is <code>true</code>

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

69

<code>allowFindMyFriendsModification</code>	Availability: Available in iOS and in tvOS 11.3 and later. Optional. Supervised only. If set to <code>1</code> changes to Find My Friends are disabled.
<code>allowFingerprintUnlock</code>	Availability: Available only in iOS 7.0 and later. Optional. If <code>1</code> prevents Touch ID from unlocking a device.
<code>allowGameCenter</code>	Availability: Available only in iOS 7 and later and in macOS 10.12.4 and later. Optional. Supervised only. When <code>1</code> Game Center is disabled and its icon is removed from the Home screen. Default is <code>true</code>
<code>allowGlobalBackgroundFetchWhenRoaming</code>	Availability: Available only in iOS 6.0 and later. Optional. When <code>1</code> disables global background fetch activity when an iOS phone is roaming.
<code>allowInAppPurchasing</code>	Availability: Available only in iOS 7 and later. Optional. When <code>1</code> prohibits in-app purchasing.
<code>allowLockScreen</code>	Availability: Available only in iOS 7 and later. Optional. If <code>1</code> prevents Control Center from appearing on the Lock screen.
<code>allowHostPairing</code>	Availability: Available only in iOS 7 and later. Supervised only. If set to <code>1</code> host pairing is disabled with the exception of the supervision host. If no supervision host certificate has been configured, all pairing is disabled. Host pairing lets the administrator control which devices an iOS 7 device can pair with.
<code>allowLockScreenNotificationsView</code>	Availability: Available only in iOS 7.0 and later. Optional. If set to <code>1</code> Notifications history view on the lock screen is disabled and users can't view past notifications. Though, when the device is locked, the user will still be able to view notifications when they arrive.
<code>allowLockScreenView</code>	Availability: Available only in iOS 7.0 and later. Optional. If set to <code>1</code> Today view in Notification Center on the lock screen is disabled.
<code>allowMultiplayer</code>	Availability: Available only in iOS 7.0 and later. Optional. When <code>1</code> prohibits multiplayer gaming. This key is deprecated on unsupervised devices.
<code>allowOpenFromManagedToUnmanaged</code>	Availability: Available only in iOS 7.0 and later. Optional. If <code>1</code> documents in managed apps and accounts only open in other managed apps and accounts. Default is <code>true</code>
<code>allowOpenFromUnmanagedToManaged</code>	Availability: Available only in iOS 7.0 and later. Optional. If set to <code>1</code> documents in unmanaged apps and accounts will only open in other unmanaged apps and accounts. Default is <code>true</code>

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

69

Key	Type	Value
<code>allowOTAPKIUpdate</code>	Boolean	Optional. If <code>1</code> over-the-air PKI updates are disabled. Setting this restriction to <code>1</code> does not disable CRL and OCSP checks. Default is <code>true</code>
Availability:		Available only in iOS 7.0 and later.
<code>allowPassbookWhenLocked</code>	Boolean	Optional. If set to <code>1</code> Passbook notifications will not be shown on the lock screen. This will default to <code>true</code>
Availability:		Available in iOS 6.0 and later.
<code>allowPhotoStream</code>	Boolean	Optional. When <code>1</code> disables Photo Stream.
Availability:		Available in iOS 5.0 and later.
<code>allowSafari</code>	Boolean	Optional. When <code>1</code> Safari web browser application is disabled and its icon removed from the Home screen. This also prevents users from opening web clips. This key is deprecated on unsupervised devices.
<code>safariAllowAutoFill</code>	Boolean	Optional. When <code>1</code> Safari auto-fill is disabled. Defaults to <code>true</code>
<code>safariForceFraudWarning</code>	Boolean	Optional. When <code>true</code> , Safari fraud warning is enabled. Defaults to <code>false</code>
<code>safariAllowJavaScript</code>	Boolean	Optional. When <code>1</code> Safari will not execute JavaScript. Defaults to <code>true</code>
<code>safariAllowPopups</code>	Boolean	Optional. When <code>1</code> Safari will not allow pop-up tabs. Defaults to <code>true</code>
<code>safariAcceptCookies</code>	Boolean	Optional. Determines conditions under which the device will accept cookies. The user facing settings changed in iOS 11, though the possible values remain the same: <ul style="list-style-type: none"> 0: Prevent Cross-Site Tracking and Block All Cookies are enabled and the user can't disable either setting. 1 or 1.5: Prevent Cross-Site Tracking is enabled and the user can't disable it. Block All Cookies is not enabled, though the user can enable it. 2: Prevent Cross-Site Tracking is enabled and Block All Cookies is not enabled. The user can toggle either setting. (Default) These are the allowed values and settings in iOS 10 and earlier: <ul style="list-style-type: none"> 0: Never 1: Allow from current website only 1.5: Allow from websites visited (Available in iOS 8.0 and later); enter '<code><real>1.5</real></code>' 2: Always (Default) In iOS 10 and earlier, users can always pick an option that is more restrictive than the payload policy, but not a less restrictive policy. For example, with a payload value of 1.5, a user could switch to <code>Never</code> , but not <code>Always Allow</code> .

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

70

Key	Type	Value
<code>allowOTAPKIUpdate</code>	Boolean	Optional. If <code>1</code> over-the-air PKI updates are disabled. Setting this restriction to <code>1</code> does not disable CRL and OCSP checks. Default is <code>true</code>
Availability:		Available only in iOS 7.0 and later.
<code>allowPassbookWhenLocked</code>	Boolean	Optional. If set to <code>1</code> Passbook notifications will not be shown on the lock screen. This will default to <code>true</code>
Availability:		Available in iOS 6.0 and later.
<code>allowPhotoStream</code>	Boolean	Optional. When <code>1</code> disables Photo Stream.
Availability:		Available in iOS 5.0 and later.
<code>allowSafari</code>	Boolean	Optional. When <code>1</code> Safari web browser application is disabled and its icon removed from the Home screen. This also prevents users from opening web clips. This key is deprecated on unsupervised devices.
<code>safariAllowAutoFill</code>	Boolean	Optional. When <code>1</code> Safari auto-fill is disabled. Defaults to <code>true</code>
<code>safariForceFraudWarning</code>	Boolean	Optional. When <code>true</code> , Safari fraud warning is enabled. Defaults to <code>false</code>
<code>safariAllowJavaScript</code>	Boolean	Optional. When <code>1</code> Safari will not execute JavaScript. Defaults to <code>true</code>
<code>safariAllowPopups</code>	Boolean	Optional. When <code>1</code> Safari will not allow pop-up tabs. Defaults to <code>true</code>
<code>safariAcceptCookies</code>	Boolean	Optional. Determines conditions under which the device will accept cookies.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

70

Key	Type	Value
<code>safariAcceptCookies</code>	Boolean	Optional. Determines conditions under which the device will accept cookies.

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.
allowUIConfigurationProfileInstallation	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.
allowUntrustedTLS	Boolean	Optional. When false, automatically rejects untrusted HTTPS certificates without prompting the user. Availability: Available in iOS 5.0 and later.
allowVideoConferencing	Boolean	Optional. When false, disables video conferencing. This key is deprecated on unsupervised devices.
allowVoiceDialing	Boolean	Optional. When false, disables voice dialing if the device is locked with a passcode. Default is true.
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, iTunes Music Store is disabled and its icon is removed from the Home screen. Users cannot preview, purchase, or download content. This key is deprecated on unsupervised devices.
allowiTunesFileSharing	Boolean	Optional. When false, iTunes application file sharing services are disabled. Availability: Available in macOS 10.13 and later.
autonomousSingleAPM	Optional	Supervised only. If present, allows apps identified by PermittedAppIDs Strings to automatically enter Single App Mode. Availability: Available only in iOS 7.0 and later.
forceAssistantFilter	Boolean	Optional. Supervised only. When true, forces the use of the profanity filter assistant.
forceEncryptedBackup	Boolean	Optional. When true, encrypts all backups.
forceiTunesStorePasswordEntry	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.
forceAirPlayOutRequestsPairingPassword	Boolean	Optional. If set to true, forces all devices receiving AirPlay RequestsPairingPassword from this device to use a pairing password. Default is false. Availability: Available only in iOS 7.1 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

71

Key	Type	Value
forceAirPlayInRequestsPairingPassword	Boolean	Optional. If set to true, forces all devices sending AirPlay RequestsPairingPassword to this device to use a pairing password. Default is false. Availability: Available only in Apple TV 6.1 to tvOS 10.1. This is not supported as of tvOS 10.2. It is recommended to use the AirPlay Security Payload.
allowManagedAppSync	Boolean	Optional. If set to false, prevents managed applications from using iCloud sync.
allowEraseContentSettings	Boolean	Supervised only. If set to false, disables the "Erase All Content And Settings" option in the Reset UI.
allowSpotlightInternetResults	Boolean	Supervised only. If set to false, Spotlight will not return Internet search results. Availability: Available in iOS and in macOS 10.11 and later.
allowEnablingRestrictions	Boolean	Supervised only. If set to false, disables the "Enable Restrictions" option in the Restrictions UI in Settings. Default is true. On iOS 12 or later, if set to false, disables the "Enable ScreenTime" option in the ScreenTime UI in Settings and disables ScreenTime if already enabled. Availability: Available in iOS 8.0 and later.
allowActivityContinuation	Boolean	If set to false, Activity Continuation will be disabled. Defaults to true.
allowEnterpriseBackup	Boolean	If set to false, Enterprise books will not be backed up. Defaults to true.
allowEnterpriseMetadataSync	Boolean	If set to false, Enterprise books notes and highlights will not be synced. Defaults to true.
allowPodcasts	Boolean	Supervised only. If set to false, disables podcasts. Defaults to true. Availability: Available in iOS 8.0 and later.
allowDefinitionLookup	Boolean	Supervised only. If set to false, disables definition lookup. Defaults to true. Availability: Available in iOS 8.1.3 and later and in macOS 10.11.2 and later.
allowPredictiveKeyboard	Boolean	Supervised only. If set to false, disables predictive keyboards.

i-net PDFC comparison results from 26/03/2019

The user facing settings changed in iOS 11, though the possible values remain the same:

- 0: Prevent Cross-Site Tracking and Block All Cookies are enabled and the user can't disable either setting.
- 1 or 1.5: Prevent Cross-Site Tracking is enabled and the user can't disable it. Block All Cookies is not enabled, though the user can enable it.
- 2: Prevent Cross-Site Tracking is enabled and Block All Cookies is not enabled. The user can toggle either setting. (Default)

These are the allowed values and settings in iOS 10 and earlier:

- 0: Never
- 1: Allow from current website only
- 1.5: Allow from websites visited (Available in iOS 8.0 and later); enter '<real>1.5</real>'
- 2: Always (Default)

In iOS 10 and earlier, users can always pick an option that is more restrictive than the payload policy, but not a less restrictive policy. For example, with a payload value of 1.5, a user could switch to Never, but not Always Allow.

allowSharedStream Available in iOS 4.0 and later.

Optional. If set to false, Shared Photo Stream will be disabled. This will default to true.
Availability: Available in iOS 6.0 and later.

allowUIConfigurationProfileInstallation

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.

allowUntrustedTLS Optional.

When false, automatically rejects untrusted HTTPS certificates without prompting the user.

Availability: Available in iOS 5.0 and later.

allowVideoConferencing Optional. When false, disables video conferencing. This key is deprecated on unsupervised devices.

allowVoiceDialing Optional. When false, disables voice dialing if the device is locked with a passcode. Default is true.

allowYouTube 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 Optional. When false, iTunes Music Store is disabled and its icon is removed from the Home screen. Users cannot preview, purchase, or download content. This key is deprecated on unsupervised devices.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

71

Key	Type	Value
allowiTunesFileSharing	Boolean	Optional. When false, iTunes application file sharing services are disabled. Availability: Available in macOS 10.13 and later.
autonomousSingleAPM	Optional	Supervised only. If present, allows apps identified by PermittedAppIDs Strings to automatically enter Single App Mode. Availability: Available only in iOS 7.0 and later.
forceAssistantFilter	Boolean	Optional. Supervised only. When true, forces the use of the profanity filter assistant.
forceEncryptedBackup	Boolean	Optional. When true, encrypts all backups.
forceiTunesStorePasswordEntry	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.
forceAirPlayOutRequestsPairingPassword	Boolean	Optional. If set to true, forces all devices receiving AirPlay RequestsPairingPassword from this device to use a pairing password. Default is false. Availability: Available only in iOS 7.1 and later.
forceAirPlayInRequestsPairingPassword	Boolean	Optional. If set to true, forces all devices sending AirPlay RequestsPairingPassword to this device to use a pairing password. Default is false. Availability: Available only in Apple TV 6.1 to tvOS 10.1. This is not supported as of tvOS 10.2. It is recommended to use the AirPlay Security Payload.
allowManagedAppSync	Boolean	Optional. If set to false, prevents managed applications from using iCloud sync.
allowEraseContentSettings	Boolean	Supervised only. If set to false, disables the "Erase All Content And Settings" option in the Reset UI.
allowSpotlightInternetResults	Boolean	Supervised only. If set to false, Spotlight will not return Internet search results. Availability: Available in iOS and in macOS 10.11 and later.
allowEnablingRestrictions	Boolean	Supervised only. If set to false, disables the "Enable Restrictions" option in the Restrictions UI in Settings. Default is true. On iOS 12 or later, if set to false, disables the "Enable ScreenTime" option in the ScreenTime UI in Settings and disables ScreenTime if already enabled. Availability: Available in iOS 8.0 and later.
allowActivityContinuation	Boolean	If set to false, Activity Continuation will be disabled. Defaults to true.
allowEnterpriseBackup	Boolean	If set to false, Enterprise books will not be backed up. Defaults to true.
allowEnterpriseMetadataSync	Boolean	If set to false, Enterprise books notes and highlights will not be synced. Defaults to true.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

72

allowAutoCorrection	Boolean	Defaults to true. Availability: Available in iOS 8.1.3 and later.
allowSpellCheck	Boolean	Supervised only. If set to false disables keyboard auto-correction. Defaults to true. Availability: Available in iOS 8.1.3 and later.
forceWatchWristDetection	Boolean	Supervised only. If set to false disables keyboard spell-check. Defaults to true. Availability: Available in iOS 8.1.3 and later.
forceWatchWristDetection	Boolean	If set to true a paired Apple Watch will be forced to use Wrist Detection. Defaults to false. Availability: Available in iOS 8.2 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

72

Key	Type	Value
allowMusicService	Boolean	Supervised only. If set to false Music service is disabled and Music app reverts to classic mode. Defaults to true. Availability: Available in iOS 9.3 and later and macOS 10.12 and later.
allowCloudPhotoLibrary	Boolean	If set to false disables iCloud Photo Library. Any photos not fully downloaded from iCloud Photo Library to the device will be removed from local storage. Availability: Available in iOS 9.0 and later and in macOS 10.12 and later.
allowNews	Boolean	Supervised only. If set to false disables News. Defaults to true. Availability: Available in iOS 9.0 and later.
forceAirDropUnmanaged	Boolean	Optional. If set to true causes AirDrop to be considered an unmanaged drop target. Defaults to false. Availability: Available in iOS 9.0 and later.
allowUIAppInstall	Boolean	Supervised only. When false App Store is disabled and its icon is removed from the Home screen. However, users may continue to use Host apps (iTunes, Configurator) to install or update their apps. Defaults to true. In iOS 10 and later, MDM commands can override this restriction. Availability: Available in iOS 9.0 and later.
allowScreenshot	Boolean	Optional. If set to false users can't save a screenshot of the display and are prevented from capturing a screen recording; it also prevents the Classroom app from observing remote screens. Defaults to true. Availability: Updated in iOS 9.0 to include screen recordings.
allowKeyboardShortcuts	Boolean	If set to false keyboard shortcuts cannot be used. Defaults to true. Availability: Available in iOS 9.0 and later.
allowPairedWatch	Boolean	Supervised only. If set to false disables pairing with an Apple Watch. Any currently paired Apple Watch is unpaired and erased. Defaults to true. Availability: Available in iOS 9.0 and later.
allowPasscodeModification	Boolean	Supervised only. If set to false prevents the device passcode from being added, changed, or removed. Defaults to true. This restriction is ignored by shared iPads. Availability: Available in iOS 9.0 and later.
allowDeviceNameModification	Boolean	Supervised only. If set to false prevents device name from being changed. Defaults to true. Availability: Available in iOS 9.0 and tvOS 11.0 and later.
allowWallpaperModification	Boolean	Supervised only. If set to false prevents wallpaper from being changed. Defaults to true. Availability: Available in iOS 9.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

73

Key	Type	Value
allowAutomaticAppDownloads	Boolean	Supervised only. If set to false prevents automatic downloading of apps purchased on other devices. Does not affect updates to existing apps. Defaults to true. Availability: Available in iOS 9.0 and later.
allowEnterpriseApp	Boolean	If set to false removes the Trust Enterprise Developer button in Settings->General->Profiles & Device Management, preventing apps from being provisioned by universal provisioning profiles. This restriction applies to free developer accounts but it does not apply to enterprise app developers who are trusted because their apps were pushed via MDM, nor does it revoke previously granted trust. Defaults to true. Availability: Available in iOS 9.0 and later.
allowRadioService	Boolean	Supervised only. If set to false Apple Music Radio is disabled. Defaults to true. Availability: Available in iOS 9.3 and later.
blacklistedAppBundleArray	String	Supervised only. If present, prevents bundle IDs listed in the array from being shown or launchable. Include the value com.apple.webscips. Availability: Available in iOS 9.3 and later.
whitelistedAppBundleArray	String	Supervised only. If present, allows only bundle IDs listed in the array from being shown or launchable. Include the value com.apple.webscips. Availability: Available in iOS 9.3 and later.
allowNotificationModification	Boolean	Supervised only. If set to false notification settings cannot be modified. Defaults to true. Availability: Available in iOS 9.3 and later.
allowRemoteScreenObservation	Boolean	If set to false prevents remote screen observation by the Classroom app is disabled. Defaults to true. This key should be nested beneath a allowScreenShot sub-restriction. If a allowScreenShot is false prevents the Classroom app from observing remote screens.

Key	Type	Value
allowPodcasts	Boolean	Supervised only. If set to false disables podcasts. Defaults to true. Availability: Available in iOS 8.0 and later.
allowDefinitionLookup	Boolean	Supervised only. If set to false disables definition lookup. Defaults to true. Availability: Available in iOS 8.1.3 and later and in macOS 10.11.2 and later.
allowPredictiveKeyboard	Boolean	Supervised only. If set to false disables predictive keyboards. Defaults to true. Availability: Available in iOS 8.1.3 and later.
allowAutoCorrection	Boolean	Supervised only. If set to false disables keyboard auto-correction. Defaults to true. Availability: Available in iOS 8.1.3 and later.
allowSpellCheck	Boolean	Supervised only. If set to false disables keyboard spell-check. Defaults to true. Availability: Available in iOS 8.1.3 and later.
forceWatchWristDetection	Boolean	If set to true a paired Apple Watch will be forced to use Wrist Detection. Defaults to false. Availability: Available in iOS 8.2 and later.
allowMusicService	Boolean	Supervised only. If set to false Music service is disabled and Music app reverts to classic mode. Defaults to true. Availability: Available in iOS 9.3 and later and macOS 10.12 and later.
allowCloudPhotoLibrary	Boolean	If set to false disables iCloud Photo Library. Any photos not fully downloaded from iCloud Photo Library to the device will be removed from local storage. Availability: Available in iOS 9.0 and later and in macOS 10.12 and later.
allowNews	Boolean	Supervised only. If set to false disables News. Defaults to true. Availability: Available in iOS 9.0 and later.
forceAirDropUnmanaged	Boolean	Optional. If set to true causes AirDrop to be considered an unmanaged drop target. Defaults to false. Availability: Available in iOS 9.0 and later.
allowUIAppInstall	Boolean	Supervised only. When false App Store is disabled and its icon is removed from the Home screen. However, users may continue to use Host apps (iTunes, Configurator) to install or update their apps. Defaults to true. In iOS 10 and later, MDM commands can override this restriction. Availability: Available in iOS 9.0 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

73

Key	Type	Value
allowScreenshot	Boolean	Optional. If set to false users can't save a screenshot of the display and are prevented from capturing a screen recording; it also prevents the Classroom app from observing remote screens. Defaults to true. Availability: Updated in iOS 9.0 to include screen recordings. Available in macOS 10.14.4 and later.
allowKeyboardShortcuts	Boolean	If set to false keyboard shortcuts cannot be used. Defaults to true. Availability: Available in iOS 9.0 and later.
allowPairedWatch	Boolean	Supervised only. If set to false disables pairing with an Apple Watch. Any currently paired Apple Watch is unpaired and erased. Defaults to true. Availability: Available in iOS 9.0 and later.
allowPasscodeModification	Boolean	Supervised only. If set to false prevents the device passcode from being added, changed, or removed. Defaults to true. This restriction is ignored by shared iPads. Availability: Available in iOS 9.0 and later.
allowDeviceNameModification	Boolean	Supervised only. If set to false prevents device name from being changed. Defaults to true. Availability: Available in iOS 9.0 and tvOS 11.0 and later.
allowWallpaperModification	Boolean	Supervised only. If set to false prevents wallpaper from being changed. Defaults to true. Availability: Available in iOS 9.0 and later.
allowAutomaticAppDownloads	Boolean	Supervised only. If set to false prevents automatic downloading of apps purchased on other devices. Does not affect updates to existing apps. Defaults to true. Availability: Available in iOS 9.0 and later.
allowEnterpriseApp	Boolean	If set to false removes the Trust Enterprise Developer button in Settings->General->Profiles & Device Management, preventing apps from being provisioned by universal provisioning profiles. This restriction applies to free developer accounts but it does not apply to enterprise app developers who are trusted because their apps were pushed via MDM, nor does it revoke previously granted trust. Defaults to true. Availability: Available in iOS 9.0 and later.
allowRadioService	Boolean	Supervised only. If set to false Apple Music Radio is disabled. Defaults to true. Availability: Available in iOS 9.3 and later.
blacklistedAppBundleArray	String	Supervised only. If present, prevents bundle IDs listed in the array from being shown or launchable. Include the value com.apple.webscips. Availability: Available in iOS 9.3 and later.
whitelistedAppBundleArray	String	Supervised only. If present, allows only bundle IDs listed in the array from being shown or launchable. Include the value com.apple.webscips. Availability: Available in iOS 9.3 and later.
allowNotificationModification	Boolean	Supervised only. If set to false notification settings cannot be modified. Defaults to true. Availability: Available in iOS 9.3 and later.
allowRemoteScreenObservation	Boolean	If set to false prevents remote screen observation by the Classroom app is disabled. Defaults to true. This key should be nested beneath a allowScreenShot sub-restriction. If a allowScreenShot is false prevents the Classroom app from observing remote screens.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

74

allowDiagnosticSubmission	Boolean	Availability: Available in iOS 9.3 and later.
		Supervised only. If set to <code>1</code> , the diagnostic submission and app analytics settings in the Diagnostics & Usage pane in Settings cannot be modified. Defaults to <code>true</code> .
allowBluetoothModification	Boolean	Availability: Available in iOS 9.3.2 and later.
		Supervised only. If set to <code>1</code> , prevents modification of Bluetooth settings. Defaults to <code>true</code> .
allowAutoUnlock	Boolean	Availability: Available in iOS 10.0 and later.
		If set to <code>1</code> , disallows macOS auto unlock. Defaults to <code>true</code> .
allowCloudDesktopDocuments	Boolean	Availability: Available only in macOS 10.12 and later.
		If set to <code>1</code> , disallows macOS cloud desktop and document services. Defaults to <code>true</code> .
		Availability: Available only in macOS 10.12.4 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

74

Key	Type	Value
allowDictation	Boolean	Supervised only. If set to <code>1</code> , disallows dictation input. Defaults to <code>true</code> .
		Availability: Available only in iOS 10.3 and later.
allowRemoteAppPairing	Boolean	If set to <code>1</code> , the Apple TV cannot be paired for use with the Remote app or Control Center widget. Defaults to <code>true</code> .
		Availability: Available in tvOS 10.2 and later.
forceWiFiWhitelisting	Boolean	Optional. Supervised only. If set to <code>true</code> , the device can join Wi-Fi networks only if they were set up through a configuration profile. Defaults to <code>false</code> .
		Availability: Available only in iOS 10.3 and later.
forceUnpromptedSignIn	Boolean	Deprecated in iOS 11. Use <code>ClassroomScreenObservationClassroomUnpromptedSignInObservation</code> instead.
allowAirPrint	Boolean	Supervised only. If set to <code>1</code> , disallows AirPrint. Defaults to <code>true</code> .
		Availability: Available in iOS 11.0 and later.
allowAirPrintCredentialsStorage	Boolean	Supervised only. If set to <code>1</code> , disallows keychain storage of username and password for Airprint. Defaults to <code>true</code> .
		Availability: Available in iOS 11.0 and later.
forceAirPrintTLSRequirement	Boolean	Supervised only. If set to <code>true</code> , requires trusted certificates for TLS printing communication. Defaults to <code>false</code> .
		Availability: Available in iOS 11.0 and later.
allowAirPrintiBaconDiscovery	Boolean	Supervised only. If set to <code>1</code> , disables iBeacon discovery of AirPrint printers. This prevents spurious AirPrint Bluetooth beacons from phishing for network traffic. Defaults to <code>true</code> .
		Availability: Available in iOS 11.0 and later.
allowCellularPlanModification	Boolean	Supervised only. If set to <code>1</code> , users can't change any settings related to their cellular plan. Defaults to <code>true</code> .
		Availability: Available in iOS 11.0 and later.
allowProximitySetupNewDevice	Boolean	Supervised only. If set to <code>1</code> , disables the prompt to setup new devices that are nearby. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowSystemAppRemoval	Boolean	Supervised only. If set to <code>1</code> , disables the removal of system apps from the device. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowVPNCreation	Boolean	Supervised only. If set to <code>1</code> , allows the creation of VPN configurations. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowUSBRestriction	Boolean	Supervised only. If set to <code>1</code> , the device will always be able to connect to USB accessories while locked. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.4.1 and later.
forceDelayedSoftwareUpdates	Boolean	Supervised only. If set to <code>true</code> , delays user visibility of Software Updates. Defaults to <code>false</code> .
		On macOS, seed build updates will be allowed, without delay.
		Availability: Available in iOS 11.3 and macOS 10.13 and tvOS 12.2 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

75

Key	Type	Value
enforcedSoftwareDelayedUpdateDelay	Boolean	Supervised only. This restriction allows the admin to set how many days a software update on the device will be delayed. With this restriction in place, the user will not see a software update until the specified number of days after the software update release date.
		The max is 90 days and the default value is 30.
		Availability: Available in iOS 11.3 and macOS 10.13.4 and tvOS 12.2 later.
forceAuthenticationBeforeAutoFill	Boolean	Optional. Supervised only. If set to <code>true</code> , the user will have to authenticate before passwords or credit card information can be autofilled in Safari and Apps. If this restriction is not enforced, the user can toggle this feature in settings.
		Only supported on devices with FaceID or TouchID.
		Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
forceClassroomAutomaticallyJoinClass	Boolean	Optional. Supervised only. If set to <code>true</code> , automatically give permission to the teacher's requests without prompting the student. Defaults to <code>false</code> .
		Availability: Available only in iOS 11.0 and later.
forceClassroomRequests	Boolean	Optional. Supervised only. If set to <code>true</code> , a student enrolled in an <code>PermissionToLeaveClass</code> unmanaged course via Classroom will request permission from the teacher when attempting to leave the course. Defaults to <code>false</code> .
		Availability: Available only in iOS 11.3 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

76

Key	Type	Value
whitelistedAppBundle	Array of Strings	Supervised only. If present, allows only bundle IDs listed in the array from being shown or launchable. Include the value <code>com.apple.mobilewebclips</code> .
		Availability: Available in iOS 9.3 and later.
allowNotificationModification	Boolean	Supervised only. If set to <code>1</code> , notification settings cannot be modified. Defaults to <code>true</code> .
allowRemoteScreenObservation	Boolean	If set to <code>1</code> , remote screen observation by the Classroom app is disabled. Defaults to <code>true</code> .
		This key should be nested beneath <code>allowScreenObservation</code> sub-restriction. If <code>allowScreenObservation</code> is also present, prevents the Classroom app from observing remote screens.
allowDiagnosticSubmissionModification	Boolean	Supervised only. If set to <code>1</code> , diagnostic submission and app analytics settings in the Diagnostics & Usage pane in Settings cannot be modified. Defaults to <code>true</code> .
allowBluetoothModification	Boolean	Supervised only. If set to <code>1</code> , prevents modification of Bluetooth settings. Defaults to <code>true</code> .
allowAutoUnlock	Boolean	If set to <code>1</code> , disallows macOS auto unlock. Defaults to <code>true</code> .
allowCloudDesktopDocuments	Boolean	If set to <code>1</code> , disallows macOS cloud desktop and document services. Defaults to <code>true</code> .
allowDictation	Boolean	Supervised only. If set to <code>1</code> , disallows dictation input. Defaults to <code>true</code> .
allowRemoteAppPairing	Boolean	If set to <code>1</code> , the Apple TV cannot be paired for use with the Remote app or Control Center widget. Defaults to <code>true</code> .
forceWiFiWhitelisting	Boolean	Optional. Supervised only. If set to <code>true</code> , the device can join Wi-Fi networks only if they were set up through a configuration profile. Defaults to <code>false</code> .
		Availability: Available only in iOS 10.12.4 and later.
forceUnpromptedSignIn	Boolean	Deprecated in iOS 11. Use <code>ClassroomScreenObservationClassroomUnpromptedSignInObservation</code> instead.
allowAirPrint	Boolean	Supervised only. If set to <code>1</code> , disallows AirPrint. Defaults to <code>true</code> .
		Availability: Available only in macOS 10.12 and later.
allowAirPrintCredentialsStorage	Boolean	Supervised only. If set to <code>1</code> , disallows keychain storage of username and password for Airprint. Defaults to <code>true</code> .
		Availability: Available only in iOS 10.12.4 and later.
forceAirPrintTLSRequirement	Boolean	Supervised only. If set to <code>true</code> , requires trusted certificates for TLS printing communication. Defaults to <code>false</code> .
		Availability: Available in iOS 11.0 and later.
allowAirPrintiBaconDiscovery	Boolean	Supervised only. If set to <code>1</code> , disables iBeacon discovery of AirPrint printers. This prevents spurious AirPrint Bluetooth beacons from phishing for network traffic. Defaults to <code>true</code> .
		Availability: Available in tvOS 10.2 and later.
forceUnpromptedSignInObservation	Boolean	Deprecated in iOS 11. Use <code>ClassroomScreenObservationClassroomUnpromptedSignInObservation</code> instead.
allowAirPrint	Boolean	Supervised only. If set to <code>1</code> , disallows AirPrint. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowAirPrintCredentialsStorage	Boolean	Supervised only. If set to <code>1</code> , disallows keychain storage of username and password for Airprint. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

75

Key	Type	Value
forceAirPrintTLSRequirement	Boolean	Supervised only. If set to <code>true</code> , requires trusted certificates for TLS printing communication. Defaults to <code>false</code> .
		Availability: Available in iOS 11.0 and later.
allowAirPrintiBaconDiscovery	Boolean	Supervised only. If set to <code>1</code> , disables iBeacon discovery of AirPrint printers. This prevents spurious AirPrint Bluetooth beacons from phishing for network traffic. Defaults to <code>true</code> .
		Availability: Available in iOS 11.0 and later.
allowCellularPlanModification	Boolean	Supervised only. If set to <code>1</code> , users can't change any settings related to their cellular plan. Defaults to <code>true</code> .
		Availability: Available in iOS 11.0 and later.
allowProximitySetupNewDevice	Boolean	Supervised only. If set to <code>1</code> , disables the prompt to setup new devices that are nearby. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowSystemAppRemoval	Boolean	Supervised only. If set to <code>1</code> , disables the removal of system apps from the device. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowVPNCreation	Boolean	Supervised only. If set to <code>1</code> , allows the creation of VPN configurations. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
allowUSBRestriction	Boolean	Supervised only. If set to <code>1</code> , the device will always be able to connect to USB accessories while locked. Defaults to <code>true</code> .
		Availability: Available only in iOS 11.4.1 and later.
forceDelayedSoftwareUpdates	Boolean	Supervised only. If set to <code>true</code> , delays user visibility of Software Updates. Defaults to <code>false</code> .
		On macOS, seed build updates will be allowed, without delay.
		Availability: Available in iOS 11.3 and macOS 10.13 and tvOS 12.2 and later.
enforcedSoftwareDelayedUpdateDelay	Boolean	Supervised only. This restriction allows the admin to set how many days a software update on the device will be delayed. With this restriction in place, the user will not see a software update until the specified number of days after the software update release date.
		The max is 90 days and the default value is 30.
		Availability: Available in iOS 11.3 and macOS 10.13.4 and tvOS 12.2 later.
forceAuthenticationBeforeAutoFill	Boolean	Optional. Supervised only. If set to <code>true</code> , the user will have to authenticate before passwords or credit card information can be autofilled in Safari and Apps. If this restriction is not enforced, the user can toggle this feature in settings.
		Only supported on devices with FaceID or TouchID.
		Defaults to <code>true</code> .
		Availability: Available only in iOS 11.0 and later.
forceClassroomAutomaticallyJoinClass	Boolean	Optional. Supervised only. If set to <code>true</code> , automatically give permission to the teacher's requests without prompting the student. Defaults to <code>false</code> .
		Availability: Available only in iOS 11.0 and later.
forceClassroomRequests	Boolean	Optional. Supervised only. If set to <code>true</code> , a student enrolled in an <code>PermissionToLeaveClass</code> unmanaged course via Classroom will request permission from the teacher when attempting to leave the course. Defaults to <code>false</code> .
		Availability: Available only in iOS 11.3 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

76

forceClassroomUrBpteam p	Optional. Supervised only. If set to true, allow the teacher to lock apps or the device without prompting the student. Defaults to false.
Availability: Available only in iOS 11.0 and later.	
forceClassroomUrBpteam p	Optional. Supervised only. If set to true, and ScreenObservationPermissionModification also true in the Education payload, a student enrolled in a managed course via the Classroom app will automatically give permission to that course's teacher's requests to observe the student's screen without prompting the student. Defaults to false.

Availability: Available only in iOS 11.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

76

Key	Type	Value
forceClassroom	Boolean	Optional. Supervised only. If set to true, automatically give AutomaticallyJoinClass permission to the teacher's requests without prompting the student. Defaults to false.
Availability: Available only in iOS 11.0 and macOS 10.14.4 and later.		
forceClassroomUrBpteam p	Optional. Supervised only. If set to true, a student enrolled in an PermissionToLeaveClass unmanaged course via Classroom will request permission from the teacher when attempting to leave the course. Defaults to false.	
Availability: Available only in iOS 11.3 and macOS 10.14.4 and later.		
forceClassroomUrBpteam p	Optional. Supervised only. If set to true, allow the teacher to lock apps or the device without prompting the student. Defaults to false.	
Availability: Available only in iOS 11.0 and macOS 10.14.4 and later.		
forceClassroomUrBpteam p	Optional. Supervised only. If set to true, and ScreenObservationPermissionModification also true in the Education payload, a student enrolled in a managed course via the Classroom app will automatically give permission to that course's teacher's requests to observe the student's screen without prompting the student. Defaults to false.	
Availability: Available only in iOS 11.0 and macOS 10.14.4 and later.		
ratingRegion	String	This 2-letter key is used by profile tools to display the proper ratings for given region. It is not recognized or reported by the client.
Possible values:		
<ul style="list-style-type: none"> • au: Australia • ca: Canada • fr: France • de: Germany • ie: Ireland • jp: Japan • nz: New Zealand • gb: United Kingdom • us: United States 		
Availability: Available in iOS and tvOS 11.3 and later.		
ratingMovies	Integer	This value defines the maximum level of movie content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 500: NC-17 • 400: R • 300: PG-13 • 200: PG • 100: G • 0: None 		
Availability: Available only in iOS and tvOS 11.3 and later.		
ratingTVShows	Integer	This value defines the maximum level of TV content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 600: TV-MA • 500: TV-14 • 400: TV-PG • 300: TV-G • 200: TV-Y7 • 100: TV-Y • 0: None 		
Availability: Available only in iOS and tvOS 11.3 and later.		

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

77

Key	Type	Value
ratingRegion	String	This 2-letter key is used by profile tools to display the proper ratings for given region. It is not recognized or reported by the client.
Possible values:		
<ul style="list-style-type: none"> • au: Australia • ca: Canada • fr: France • de: Germany • ie: Ireland • jp: Japan • nz: New Zealand • gb: United Kingdom • us: United States 		
Availability: Available in iOS and tvOS 11.3 and later.		
ratingMovies	Integer	This value defines the maximum level of movie content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 500: NC-17 • 400: R • 300: PG-13 • 200: PG • 100: G • 0: None 		
Availability: Available only in iOS and tvOS 11.3 and later.		
ratingTVShows	Integer	This value defines the maximum level of TV content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 600: TV-MA • 500: TV-14 • 400: TV-PG • 300: TV-G • 200: TV-Y7 • 100: TV-Y • 0: None 		
Availability: Available only in iOS and tvOS 11.3 and later.		

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

77

Key	Type	Value
ratingMovies	Integer	This value defines the maximum level of movie content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 500: NC-17 • 400: R • 300: PG-13 • 200: PG • 100: G • 0: None 		
Availability: Available only in iOS and tvOS 11.3 and later.		
ratingTVShows	Integer	This value defines the maximum level of TV content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 600: TV-MA • 500: TV-14 • 400: TV-PG • 300: TV-G • 200: TV-Y7 • 100: TV-Y • 0: None 		
Availability: Available only in iOS and tvOS 11.3 and later.		
ratingApps	Integer	This value defines the maximum level of app content that is allowed on the device.
Possible values (with the US description of the rating level):		
<ul style="list-style-type: none"> • 1000: All • 600: 17+ • 300: 12+ • 200: 9+ • 100: 4+ • 0: None 		
Availability: Available only in iOS 5 and tvOS 11.3 and later.		
forceAutomaticDateAndTime	Boolean	Optional. Supervised only. If set to true, the Date & Time "Set Automatically" feature is turned on and can't be turned off by the user. Defaults to false.
Note: The device's time zone will only be updated when the device can determine its location (cellular connection or wifi with location services enabled).		
Availability: Available only in iOS 12.0 and tvOS 12.2 and later.		
allow>PasswordAutofill	Boolean	Optional. Supervised only. If set to true, users will not be able to use the Autofill Passwords feature on iOS and will not be prompted to use a saved password in Safari or in apps. If set to false, automatic Strong Passwords will also be disabled and

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

strong passwords will not be suggested to users. Defaults to true.
Availability: Available only in iOS 12.0 and macOS 10.14 and later.
allowPasswordRequests Optional. Supervised only. If set to false a user's device will not request passwords from nearby devices. Defaults to true.
Availability: Available only in iOS 12.0, macOS 10.14, and tvOS 12.0 and later.
allowPasswordShares Optional. Supervised only. If set to false users can not share their passwords with the Airdrop Passwords feature. Defaults to true.
Availability: Available only in iOS 12.0 and macOS 10.14 and later.
allowManagedToUnmanagedContacts Optional. If set to true managed apps can write contacts to unmanaged contacts accounts. Defaults to false if a low OpenFromManagedToUnmanaged restriction has no effect.
A payload that sets this to true must be installed via MDM.
Availability: Available only in iOS 12.0 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

78

Key	Type	Value
allowUnmanagedToManagedContacts	Optional. If set to true unmanaged apps can read from managed contacts accounts. Defaults to false if a low OpenFromManagedToUnmanaged restriction has no effect.	
A payload that sets this to true must be installed via MDM.	Availability: Available only in iOS 12.0 and later.	
allowESIMModifications	Optional. Supervised only. If set to false the user may not remove or add a cellular plan to the eSIM on the device. Defaults to true.	
Availability: Available only in iOS 12.1 and later.		
allowPersonalHotspotModification	Optional. Supervised only. If set to false the user may not modify the personal hotspot setting. Defaults to true.	
Availability: Available only in iOS 12.2 and later.		
allowSiriServerLogging	Optional. If set to false server-side Siri logging is disabled. Defaults to true.	
Availability: Available only in iOS 12.2 and later.		

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

79

SCEP Payload

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

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

Key	Type	Value
<code>URL</code>	String	The SCEP URL. See Over-the-Air Profile Delivery and Configuration for more information about SCEP.
<code>Name</code>	String	Optional. Any string that is understood by the SCEP server. For example, it could be a domain name like <code>example.com</code> . If a certificate authority has multiple CA certificates this field can be used to distinguish which is required.
<code>Subject</code>	Array	Optional. The representation of a X.500 name represented as an array of OID and value. For example, <code>/C=US/O=Apple Inc./CN=foo/1</code> which would translate to: <code>[["C", "US"], ["O", "Apple Inc."], ["1.2.5.3", "bar"]]</code> OIDs can be represented as dotted numbers, with shortcuts for country (C), locality (L), state (S), organization (O), organizational unit (OU), and common name (CN).

Key	Type	Value
<code>allowPasswordAutofill</code>	Optional. Supervised only. If set to false a user will not be able to use the AutoFill Passwords feature on iOS and will not be prompted to use a saved password in Safari or in apps. If set to false Automatic Strong Passwords will also be disabled and strong passwords will not be suggested to users. Defaults to true.	
Availability: Available only in iOS 12.0 and macOS 10.14 and later.		
<code>allowPasswordRequests</code>	Optional. Supervised only. If set to false a user's device will not request passwords from nearby devices. Defaults to true.	
Availability: Available only in iOS 12.0, macOS 10.14, and tvOS 12.0 and later.		
<code>allowPasswordShares</code>	Optional. Supervised only. If set to false users can not share their passwords with the Airdrop Passwords feature. Defaults to true.	
Availability: Available only in iOS 12.0 and macOS 10.14 and later.		
<code>allowManagedToUnmanagedContacts</code>	Optional. If set to true managed apps can write contacts to unmanaged contacts accounts. Defaults to false if a low OpenFromManagedToUnmanaged restriction has no effect.	
A payload that sets this to true must be installed via MDM.	Availability: Available only in iOS 12.0 and later.	
<code>allowUnmanagedToManagedContacts</code>	Optional. If set to true unmanaged apps can read from managed contacts accounts. Defaults to false if a low OpenFromUnmanagedToManaged restriction has no effect.	
A payload that sets this to true must be installed via MDM.	Availability: Available only in iOS 12.0 and later.	
<code>allowESIMModifications</code>	Optional. Supervised only. If set to false the user may not remove or add a cellular plan to the eSIM on the device. Defaults to true.	
Availability: Available only in iOS 12.1 and later.		
<code>allowPersonalHotspotModification</code>	Optional. Supervised only. If set to false the user may not modify the personal hotspot setting. Defaults to true.	
Availability: Available only in iOS 12.2 and later.		
<code>allowSiriServerLogging</code>	Optional. If set to false server-side Siri logging is disabled. Defaults to true.	
Availability: Available only in iOS 12.2 and later.		

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

79

SCEP Payload

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

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

Key	Type	Value
<code>URL</code>	String	The SCEP URL. See Over-the-Air Profile Delivery and Configuration for more information about SCEP.
<code>Name</code>	String	Optional. Any string that is understood by the SCEP server. For example, it could be a domain name like <code>example.com</code> . If a certificate authority has multiple CA certificates this field can be used to distinguish which is required.
<code>Subject</code>	Array	Optional. The representation of a X.500 name represented as an array of OID and value. For example, <code>/C=US/O=Apple Inc./CN=foo/1</code> which would translate to: <code>[["C", "US"], ["O", "Apple Inc."], ["1.2.5.3", "bar"]]</code> OIDs can be represented as dotted numbers, with shortcuts for country (C), locality (L), state (S), organization (O), organizational unit (OU), and common name (CN).
<code>Challenge</code>	String	Optional. A pre-shared secret.
<code>Keysize</code>	Integer	Optional. The key size in bits, either 1024 or 2048.
<code>KeyType</code>	String	Optional. Currently always "RSA".
<code>KeyUsage</code>	Integer	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.
<code>Availability:</code>		Available only in iOS 4 and later.
<code>Retries</code>	Integer	Optional. The number of times the device should retry if the server sends a PENDING response. Defaults to 3.
<code>RetryDelay</code>	Integer	Optional. The number of seconds to wait between subsequent retries. The first retry is attempted without this delay. Defaults to 10.
<code>CAFingerprintData</code>	String	Optional. The fingerprint of the Certificate Authority certificate.
<code>AllowAllApps</code>	Boolean	Optional. If true, all apps have access to the private key. Default is false.
<code>KeyIsExtractable</code>	Boolean	Optional. If a key is private key cannot be exported from the keychain. Default is true.

l l l "C", "U S" J J, l l "O", "Apple Inc." J J, ["1.2.5.3", "bar"]]			
OIDs can be represented as dotted numbers, with shortcuts for country (C), locality (L), state (S), organization (O), organizational unit (O_U), and common name (C_N).			
Challenge	String	Optional. A pre-shared secret.	
KeySize	Integer	Optional. The key size in bits, either 1024 or 2048.	
KeyType	String	Optional. Currently always "RSA".	
KeyUsage	Integer	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.			
Retries	Integer	Optional. The number of times the device should retry if the server sends a PENDING response. Defaults to 3.	
RetryDelay	Integer	Optional. The number of seconds to wait between subsequent retries. The first retry is attempted without this delay. Defaults to 10.	
CAFingerprintData	String	Optional. The fingerprint of the Certificate Authority certificate.	
AllowAllApps	Boolean	Optional. If true, all apps have access to the private key. Default is false.	
KeyIsExtractable	Boolean	Optional. If false, the private key cannot be exported from the keychain. Default is true.	

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

80

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](#) 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 `GetCAC` and uses the answer it gets in response. If the CA doesn't respond, the device defaults to `GetCaps` requests. For more information, read [Over-the-Air Profile Delivery and Configuration](#). This feature is not supported in macOS.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

81

Screensaver

Screensaver payloads are designated by specifying `com.apple.screensaver` as the payload type. The device level screensaver payload can be used to customize the screensaver and enable or disable the screen lock password function.

The Screensaver payload defines the following keys:

Key	Type	Value
<code>askForPassword</code>	Boolean	Optional. If true, the user will be prompted for a password when the screensaver is unlocked or stopped. When using this prompt, <code>askForPasswordMustBeProvided</code> must be provided.
<code>askForPassword</code>	Integer	Optional. Number of seconds to delay before the password will be required to unlock or stop the screen saver (the "grace period"). A value of 2147483647 (eg 0xFFFFFFFF) can be used to disable this requirement, and on 10.13, the payload is one of the only ways of disabling the feature. Note that <code>askForPassword</code> must be set to true to use this option.

i-net PDFC comparison results from 26/03/2019

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

80

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](#) 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 `GetCAC` and uses the answer it gets in response. If the CA doesn't respond, the device defaults to `GetCaps` requests. For more information, read [Over-the-Air Profile Delivery and Configuration](#). This feature is not supported in macOS.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

81

Screensaver

Screensaver payloads are designated by specifying `com.apple.screensaver` as the payload type. The device level screensaver payload can be used to customize the screensaver and enable or disable the screen lock password function.

The Screensaver payload defines the following keys:

Key	Type	Value
<code>askForPassword</code>	Boolean	Optional. If true, the user will be prompted for a password when the screensaver is unlocked or stopped. When using this prompt, <code>askForPasswordMustBeProvided</code> must be provided.
<code>askForPassword</code>	Integer	Optional. Number of seconds to delay before the password will be required to unlock or stop the screen saver (the "grace period"). A value of 2147483647 (eg 0xFFFFFFFF) can be used to disable this requirement, and on 10.13, the payload is one of the only ways of disabling the feature. Note that <code>askForPassword</code> must be set to true to use this option.
<code>loginWindowModule</code>	String	Optional. A full path to the screen saver module to be used.
<code>loginWindowModule</code>	Integer	Optional. Available in macOS 10.11 and later.
<code>loginWindowInactivity</code>	Integer	Optional. Number of seconds of inactivity before screensaver activates. (0=never activate).
<code>loginWindowInactivity</code>	Integer	Optional. Available in macOS 10.11 and later.

User level screensaver payloads are designated by specifying `com.apple.screensaver` as the payload type.

The user level screensaver settings are specific to a user, instead of the device.

The Screensaver User payload defines the following keys:

Key	Type	Value
<code>modulePath</code>	String	Optional. A full path to the screen saver module to be used.
<code>modulePath</code>	Integer	Optional. Available in macOS 10.11 and later.
<code>idleTime</code>	Integer	Optional. Number of seconds of inactivity before screensaver activates. (0=never activate).
<code>idleTime</code>	Integer	Optional. Available in macOS 10.11 and later.

<code>log in Window Mod e</code>	<code>String</code>	Optional. A full path to the screen saver module to be used.
		Availability: Available in macOS 10.13 and later.
<code>log in Window I dle</code>	<code>String</code>	Optional. Number of seconds of inactivity before screensaver activates. (0=never activate).
		Availability: Available in macOS 10.11 and later.

User level screensaver payloads are designated by specifying `com.apple.screensaver.userPayload`.

The user level screensaver settings are specific to a user, instead of the device.

The Screensaver User payload defines the following keys:

Key	Type	Value
<code>module PaString</code>	<code>Optional</code> . A full path to the screen saver module to be used.	
		Availability: Available in macOS 10.11 and later.
<code>idle Tim e</code>	<code>String</code>	Optional. Number of seconds of inactivity before screensaver activates. (0=never activate).
		Availability: Available in macOS 10.11 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

82

Setup Assistant

The Setup Assistant Payload is designated by specifying `com.apple.SetupAssistant.managedPayload`.

On macOS, this payload specifies Setup Assistant options for either the system or particular users.

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

Key	Type	Value
<code>Skip Cloud Setup</code>	<code>Boolean</code>	Optional. If true, skip the Apple ID setup window.
		Availability: Available in macOS 10.12 and later.
<code>Skip Siri Setup</code>	<code>Boolean</code>	Optional. If true, skip the Siri setup window.
		Availability: Available in macOS 10.12 and later.
<code>Skip Privacy Setup</code>	<code>Boolean</code>	Optional. If true, skip the Privacy consent window.
		Availability: Available in macOS 10.13.4 and later.
<code>Skip iCloud Storage Setup</code>	<code>Boolean</code>	Optional. If true, skip the iCloud Storage window.
		Availability: Available in macOS 10.13.4 and later.
<code>Skip True Tone</code>	<code>Boolean</code>	Optional. If true, skip the True Tone Display window.
		Availability: Available in macOS 10.13.6 and later.
<code>Skip Appearance</code>	<code>Boolean</code>	Optional. If true, skip the Choose Your Look window.
		Availability: Available in macOS 10.14 and later.

Shared Device Configuration Payload

The Shared Device Configuration Payload is designated by specifying `com.apple.sharedDeviceConfiguration` as the `Payload`.

The Shared Device Configuration Payload allows admins to specify optional text displayed on the login window and lock screen (i.e. a "If Lost, Return To" message and Asset Tag Information). It is supported on iOS 9.3 and later.

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

Key	Type	Value
<code>Asset Tag Information</code>	<code>String</code>	Optional. Asset tag information for the device, displayed on the login window and lock screen.
<code>Lock Screen Footer</code>	<code>String</code>	Optional. A footnote displayed on the login window and lock screen. Available in iOS 9.3 and later.
<code>If Lost Return To</code>	<code>String</code>	Deprecated. Use <code>Lock Screen Footer</code> instead.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

83

ShareKit Payload

MacOS 10.9 or later only. The ShareKit Payload is designated by specifying `com.apple.com.apple.ShareKit` as the `Payload`.

The ShareKit Payload specifies which ShareKit plugin can be accessed on client. Both allow and disallow lists can be specified.

This payload is deprecated as of macOS 10.12. For clients running macOS 10.13 or later, use the NSExtension Payload instead. If a profile contains both a NSExtension Payload and a ShareKit Payload, the ShareKit Payload will be ignored.

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

Key	Type	Value
<code>SHKA llowed Share Array</code>	<code>Optional</code> . List of plugin IDs that will show up in the user's Share menu. If this array exists then only these items will be permitted.	

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

82

Setup Assistant

The Setup Assistant Payload is designated by specifying `com.apple.SetupAssistant.managedPayload`.

On macOS, this payload specifies Setup Assistant options for either the system or particular users.

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

Key	Type	Value
<code>Skip Cloud Setup</code>	<code>Boolean</code>	Optional. If true, skip the Apple ID setup window.
		Availability: Available in macOS 10.12 and later.
<code>Skip Siri Setup</code>	<code>Boolean</code>	Optional. If true, skip the Siri setup window.
		Availability: Available in macOS 10.12 and later.
<code>Skip Privacy Setup</code>	<code>Boolean</code>	Optional. If true, skip the Privacy consent window.
		Availability: Available in macOS 10.13.4 and later.
<code>Skip iCloud Storage Setup</code>	<code>Boolean</code>	Optional. If true, skip the iCloud Storage window.
		Availability: Available in macOS 10.13.4 and later.
<code>Skip True Tone</code>	<code>Boolean</code>	Optional. If true, skip the True Tone Display window.
		Availability: Available in macOS 10.13.6 and later.
<code>Skip Appearance</code>	<code>Boolean</code>	Optional. If true, skip the Choose Your Look window.
		Availability: Available in macOS 10.14 and later.

Shared Device Configuration Payload

The Shared Device Configuration Payload is designated by specifying `com.apple.sharedDeviceConfiguration` as the `Payload` to contain only one payload, which must be supervised. It is not supported on the User Channel.

The Shared Device Configuration Payload allows admins to specify optional text displayed on the login window and lock screen (i.e. a "If Lost, Return To" message and Asset Tag Information). It is supported on iOS 9.3 and later.

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

Key	Type	Value
<code>Asset Tag Information</code>	<code>String</code>	Optional. Asset tag information for the device, displayed on the login window and lock screen.
<code>Lock Screen Footer</code>	<code>String</code>	Optional. A footnote displayed on the login window and lock screen. Available in iOS 9.3 and later.
<code>If Lost Return To</code>	<code>String</code>	Deprecated. Use <code>Lock Screen Footer</code> instead.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

83

ShareKit Payload

MacOS 10.9 or later only. The ShareKit Payload is designated by specifying `com.apple.com.apple.ShareKit` as the `Payload`.

The ShareKit Payload specifies which ShareKit plugin can be accessed on client. Both allow and disallow lists can be specified.

This payload is deprecated as of macOS 10.12. For clients running macOS 10.13 or later, use the NSExtension Payload instead. If a profile contains both a NSExtension Payload and a ShareKit Payload, the ShareKit Payload will be ignored.

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

Key	Type	Value
<code>SHKA llowed Share Array</code>	<code>Optional</code> . List of plugin IDs that will show up in the user's Share menu. If this array exists then only these items will be permitted.	
<code>SHKD enied Share Array</code>	<code>Optional</code> . List of plugin IDs that will not show up in the user's Share menu. This key is used only if there is no <code>SHKA llowed Share Array</code> .	

The following plugin IDs are supported by this payload:

- "com.apple.share.AirDrop": AirDrop
- "com.apple.share.Facebook": Facebook
- "com.apple.share.Twitter": Twitter
- "com.apple.share.Mail": Mail
- "com.apple.share.Messages": Messages
- "com.apple.share.Video": Video Services
- "com.apple.share.addtophoto": Photos
- "com.apple.share.addtoaperture": Aperture
- "com.apple.share.readlater": Reading List
- "com.apple.share.SinaWeibo": Sina Weibo
- "com.apple.Notes.SharingExtension": Notes
- "com.apple.reminders.RemindersShareExtension": Reminders
- "com.apple.share.Linkedin.post": LinkedIn

Strings	menu. If this array exists then only these items will be permitted.
SHKDeniedShares	Optional. List of plugin IDs that will not show up in the user's Share menu. This key is used only if there is no SHKAllowSharedServices

The following plugin IDs are supported by this payload:

- "com.apple.share.AirDrop": AirDrop
- "com.apple.share.Facebook": Facebook
- "com.apple.share.Twitter": Twitter
- "com.apple.share.Mail": Mail
- "com.apple.share.Messages": Messages
- "com.apple.share.Video": Video Services
- "com.apple.share.addtophoto": Photos
- "com.apple.share.addtoaperture": Aperture
- "com.apple.share.readlater": Reading List
- "com.apple.share.SinaWeibo": Sina Weibo
- "com.apple.Notes.SharingExtension": Notes
- "com.apple.reminders.RemindersShareExtension": Reminders
- "com.apple.share.LinkedIn.post": LinkedIn

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

84

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

84

Single Sign-On Account Payload

The Single Sign-On Account payload is designated by specifying `com.apple.security payload type`. 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 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.
PayloadCertificateStringUUID	String	Optional. The PayloadUUID of an identity certificate payload that can be used to renew the Kerberos credential without user interaction. The certificate payload must have either the <code>com.apple.security.pkcs12</code> or <code>com.apple.security.payload.type</code> . Both the Single Sign On payload and the identity certificate payload must be included in the same configuration profile.
Realm	String	The Kerberos realm name. This value should be properly capitalized.
URLPrefixMatcheArray	Strings	List of URLs prefixes that must be matched to use this account for Kerberos authentication over HTTP. Note that the URL postfixes must match as well.
AppIdentifierMa	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.

Each entry in the URLPrefixMatcheArray 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`. 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/`. With iOS 9.0 or 10, however, a single wildcard * may be used to specify all matching values. For example, `http://example.com/` will match both `http://store.example.com/` and `http://www.example.com/`.

The patterns `http://` and `https://` match all HTTP and HTTPS URLs, respectively.

The AppIdentifierMa array must contain strings that match app bundle IDs. These strings may be exact matches (`com.mycompany.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 once, at the

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

85

end of the string (`com.mycompany.*example`). When a wildcard is included, any app whose bundle ID begins with the prefix is granted access to the account.

SmartCard Settings Payload

The SmartCard Settings payload is designated by specifying `com.apple.security smartcard payload type`.

This payload controls restrictions and settings for SmartCard pairing on macOS v10.12.4 and later.

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

Key	Type	Value
UserPairing	Boolean	Optional. If a user will not get the pairing dialog, although existing pairings will still work. Default is <code>true</code> .
allowSmartCard	Boolean	Optional. If a smartcard is disabled for logins, authorizations, and screensaver unlocking. It is still allowed for other functions, such as signing emails and web access. A restart is required for a change of setting to take effect. Default is <code>true</code> .
checkCertificateTrust	Integer	Optional. Valid values are 0-3: <ul style="list-style-type: none"> 0: certificate trust check is turned off 1: certificate trust check is turned on. Standard validity check is being performed but this does not include additional revocation checks. 2: certificate trust check is turned on, plus a soft revocation check is performed. Until the certificate is explicitly rejected by CRL/OCSP, it is considered as valid. This implies that unavailable/unreachable CRL/OCSP allows this check to succeed. 3: certificate trust check is turned on, plus a hard revocation check is performed. Unless CRL/OCSP explicitly says "this certificate is OK", the certificate is considered as invalid. This is the most secure option.

oneCardPerUser

Optional. If a user can pair with only one SmartCard, although existing pairings will be allowed if already set up. Default is `false`.

enforceSmartCard

Optional. If a user can only login or authenticate with a

end of the string (`com.mycompany.*example`). When a wildcard is included, any app whose bundle ID begins with the prefix is granted access to the account.

SmartCard Settings Payload

The SmartCard Settings payload is designated by specifying `com.apple.security smartcard payload type`.

This payload controls restrictions and settings for SmartCard pairing on macOS v10.12.4 and later.

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

Key	Type	Value
UserPairing	Boolean	Optional. If a user will not get the pairing dialog, although existing pairings will still work. Default is true.
allowSmartCard	Boolean	Optional. If a SmartCard is disabled for logins, authorizations, and screensaver unlocking. It is still allowed for other functions, such as signing emails and web access. A restart is required for a change of setting to take effect. Default is true.
checkCertificate	Integer	Optional. Valid values are 0-3: <ul style="list-style-type: none"> 0: certificate trust check is turned off 1: certificate trust check is turned on. Standard validity check is being performed but this does not include additional revocation checks. 2: certificate trust check is turned on, plus a soft revocation check is performed. Until the certificate is explicitly rejected by CRL/OCSP, it is considered as valid. This implies that unavailable/unreachable CRL/OCSP allows this check to succeed. 3: certificate trust check is turned on, plus a hard revocation check is performed. Unless CRL/OCSP explicitly says "this certificate is OK", the certificate is considered as invalid. This is the most secure option.
oneCardPerUser	Boolean	Default is 0.
Optional. If a user can pair with only one SmartCard, although existing pairings will be allowed if already set up. Default is false.		
enforceSmartCard	Boolean	Optional. If a user can only login or authenticate with a SmartCard. Default is false.
tokenRemoval	Integer	Optional. If 1, when the SmartCard is removed, the screensaver will be enabled. Default is 0.
		Availability: Available in macOS 10.13.4 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

86

Software Update

The Software Update payload is designated by specifying `com.apple.SoftwareUpdate`.
In addition to the settings common to all payloads, this payload defines the following key:

Key	Type	Value
CatalogURL	String	Optional. The URL of the software update catalog.
AllowPreRelease	Boolean	Optional. If true, prerelease software can be installed on this computer. Default is true.
restrict-software	Boolean	Optional. If true, restrict app installations to admin users.
require-admin-to-install	Boolean	Optional. If this key has the same function as the key <code>restrict-store-require-admin-to-install</code> .
		Availability: Available in macOS 10.14 and later.

System Migration Payload

The System Migration payload is designated by specifying `com.apple.systemsMigration`.
System migration occurs when items are transferred to a macOS device from a Windows device by reading source and destination path pairs from plist files. This payload provides a way to customize those transfers.

This payload must be single and exist only 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 macOS 10.12.4 and later.

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

Key	Type	Value
CustomBehaviors	Array of Dictionaries	Optional. Specifies custom behavior for the context designated in each dictionary.

Each dictionary in the `CustomBehaviors` contains these keys:

Key	Type	Value
ContextString	String	Required. The context to which custom paths apply.
Paths	Array of Dictionaries	Required. The custom paths to be migrated from a source system to a target system.

Each dictionary in the `Paths` contains these keys:

Key	Type	Value
SourcePath	String	Required. The path to the migrating file or directory on the source system.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

87

Key	Type	Value
SourcePath	String	Required. If true, the source path is located within a user home directory.
TargetPath	String	Required. The path to the destination file or directory on the target system.
TargetPath	String	Required. If true, the target path is located within a user home directory.

tokenRemoval	Integer	SmartCard. Default is false.
		Optional. If 1, when the SmartCard is removed, the screensaver will be enabled. Default is 0.

Availability: Available in macOS 10.13.4 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

86

Software Update

The Software Update payload is designated by specifying `com.apple.SoftwareUpdate`.
In addition to the settings common to all payloads, this payload defines the following key:

Key	Type	Value
CatalogURL	String	Optional. The URL of the software update catalog.
AllowPreRelease	Boolean	Optional. If true, prerelease software can be installed on this computer. Default is true.
restrict-software	Boolean	Optional. If true, restrict app installations to admin users.
require-admin-to-install	Boolean	Optional. If this key has the same function as the key <code>restrict-store-require-admin-to-install</code> .
		Availability: Available in macOS 10.14 and later.

System Migration Payload

The System Migration payload is designated by specifying `com.apple.systemsMigration`.
System migration occurs when items are transferred to a macOS device from a Windows device by reading source and destination path pairs from plist files. This payload provides a way to customize those transfers.

This payload must be single and exist only 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 macOS 10.12.4 and later.

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

Key	Type	Value
CustomBehaviors	Array of Dictionaries	Optional. Specifies custom behavior for the context designated in each dictionary.

Each dictionary in the `CustomBehaviors` contains these keys:

Key	Type	Value
ContextString	String	Required. The context to which custom paths apply.
Paths	Array of Dictionaries	Required. The custom paths to be migrated from a source system to a target system.

Each dictionary in the `Paths` contains these keys:

Key	Type	Value
SourcePath	String	Required. The path to the migrating file or directory on the source system.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

87

Key	Type	Value
SourcePath	String	Required. If true, the source path is located within a user home directory.
TargetPath	String	Required. The path to the destination file or directory on the target system.
TargetPath	String	Required. If true, the target path is located within a user home directory.

System Policy Control Payload

The System Policy Control payload is designated by specifying `com.apple.systemPolicyControl`.
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 macOS v10.8 and later.

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

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	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 <code>EnableAssessment</code> is not true, this key has no effect.

System Policy Rule Payload

System Policy Control Payload

The System Policy Control payload is designated by specifying `com.apple.systempolicy.control.payloads`.

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 macOS v10.8 and later.

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

Key	Type	Value
<code>EnableAssessment</code>	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.
<code>AllowIdentifiedDevelopers</code>	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 <code>EnableAssessment</code> is set to NO, this key has no effect.

System Policy Rule Payload

The System Policy Rule payload is designated by specifying `com.apple.systempolicy.rules`.

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 `sptcd` command line tool. You should be read the manual page for `sptcd`.

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 macOS v10.8 and later.

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

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

88

Key**Type****Value**

<code>RequirementString</code>	The policy requirement. This key must follow the syntax described in Code Signing Requirement Language .
<code>Comment</code>	String
<code>ExpirationDate</code>	Optional. An expiration date for rule(s) being processed.
<code>OperationTString</code>	Optional. One of <code>operation:expiration:install</code> or <code>operation:allowDefaultToOperation:execute</code>

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"7696f2cbff7f7d43fc8f52f3cdc8fa0fcccdb4"`

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 `RequirementKey` using a payload variable of the form `$SHASHCERT_xx$` 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 = $SHASHCERT_Cert1Data$</string>`

and then provide:

`<key>Cert1Data</key>
<data>
MIIIFTDCCBDSgAwIBAgIHBHXzGzq8DANBgkqhkiG9w0BAQFADCByjELMAKGA
...
z1I6yBET5qaGhpWexEp3baLbXlcrtgufmDSUtUnImavGyw==</data>`

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

`certificate leaf = H"7696f2cbff7f7d43fc8f52f3cdc8fa0fcccdb4"`

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

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

89

System Policy Managed Payload

The System Policy Managed payload is designated by specifying `com.apple.systempolicy.managedPayloads`.

System Policy Rule Payload

The System Policy Rule payload is designated by specifying `com.apple.systempolicy.rules`.

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 `sptcd` command line tool. You should be read the manual page for `sptcd`.

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 macOS v10.8 and later.

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

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

88

Key**Type****Value**

<code>RequirementString</code>	The policy requirement. This key must follow the syntax described in Code Signing Requirement Language .
<code>Comment</code>	String
<code>ExpirationDate</code>	Optional. An expiration date for rule(s) being processed.

`OperationTString`

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"7696f2cbff7f7d43fc8f52f3cdc8fa0fcccdb4"`

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 `RequirementKey` using a payload variable of the form `$SHASHCERT_xx$` 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 = $SHASHCERT_Cert1Data$</string>`

and then provide:

`<key>Cert1Data</key>
<data>
MIIIFTDCCBDSgAwIBAgIHBHXzGzq8DANBgkqhkiG9w0BAQFADCByjELMAKGA
...
z1I6yBET5qaGhpWexEp3baLbXlcrtgufmDSUtUnImavGyw==</data>`

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

`certificate leaf = H"7696f2cbff7f7d43fc8f52f3cdc8fa0fcccdb4"`

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

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

89

System Policy Managed Payload

The System Policy Managed payload is designated by specifying `com.apple.systempolicy.managedPayloads`.

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

This payload is supported only on macOS v10.8 and later.

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

Key	Type	Value
<code>DisableOverwrite</code>	Boolean	Optional. If YES, the Finder's contextual menu item will be disabled.

TV Remote Payload

The TV Remote payload is designated by specifying `com.apple.tvRemotePayload`.

This payload is supported only on supervised devices.

This payload allows restricting the connections from the Apple TV Remote app to an Apple TV and restricting the available Apple TV devices in the Apple TV Remote app.

To lock specific Apple TVs to specific devices running Apple TV Remote app, both the Apple TVs and remote devices can be specified in the same payload.

In addition to the settings common to all payload types, the TV Remote payload defines the following keys:

Key	Type	Value
<code>AllowedRemoteDictionary</code>	Array of Dictionaries	If present, the Apple TV will only connect with the Apple TV Remote app from the devices specified.
<code>Availability</code>	String	If not present, or the list is empty, any device will be allowed to connect.

`AllowedTVs`

If present, the Apple TV Remote app will only connect to the specified

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

This payload is supported only on macOS v10.8 and later.

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

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

TV Remote Payload

The TV Remote payload is designated by specifying `com.apple.tvRemotePayload` as the `value`. This payload is supported only on supervised devices.

This payload allows restricting the connections from the Apple TV Remote app to an Apple TV and restricting the available Apple TV devices in the Apple TV Remote app.

To lock specific Apple TVs to specific devices running Apple TV Remote app, both the Apple TVs and remote devices can be specified in the same payload.

In addition to the settings common to all payload types, the TV Remote payload defines the following keys:

Key	Type	Value
AllowedDevices	Dictionary	If present, the Apple TV will only connect with the Apple TV Remote app from the devices specified. If not present, or the list is empty, any device will be allowed to connect. Availability: Available in tvOS 11.3 and later.
AllowedTVs	Dictionary	If present, the Apple TV Remote app will only connect to the specified Apple TVs. If not present, or the list is empty, the device will be able to connect to any Apple TV. Availability: Available in iOS 11.3 and later.

Each entry in the `AllowedDevices` dictionary that can contain the following key:

Key	Type	Value
RemoteDeviceString	String	The MAC address of a permitted iOS device that can control this Apple TV. Use the format "xx:xx:xx:xx:xx:xx". The field is not case sensitive. Availability: Available in tvOS 11.3 and later.

Each entry in the `AllowedTVs` dictionary that can contain the following key:

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.
90

Key	Type	Value
TVDeviceString	String	The MAC address of an Apple TV device that this iOS device is permitted to control. Use the format "xx:xx:xx:xx:xx:xx". The field is not case sensitive. Availability: Available in iOS 11.3 and later.

Time Server Payload

The Time Server payload is designated by specifying `com.apple.timeServer` as the `value`.

This payload allows devices to connect to custom time servers.

In addition to the settings common to all payload types, the Time Server payload defines the following keys:

Key	Type	Value
timeServerString	String	The ntp server to connect to. Availability: Available in macOS 10.12.4 and later.
timeZoneString	String	Time zone path location string in /usr/share/zoneinfo, for example, /"America/Denver" or "Zulu". Availability: Available in macOS 10.12.4 and later.

Dictionaries

Apple TVs.

If not present, or the list is empty, the device will be able to connect to any Apple TV.

Availability: Available in iOS 11.3 and later.

Each entry in the `AllowedDevices` dictionary that can contain the following key:

Key	Type	Value
RemoteDeviceString	String	The MAC address of a permitted iOS device that can control this Apple TV. Use the format "xx:xx:xx:xx:xx:xx". The field is not case sensitive. Availability: Available in tvOS 11.3 and later.

Each entry in the `AllowedTVs` dictionary that can contain the following key:

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

90

Key

Type

Value

TVDeviceString	String	The MAC address of an Apple TV device that this iOS device is permitted to control. Use the format "xx:xx:xx:xx:xx:xx". The field is not case sensitive. Availability: Available in iOS 11.3 and later.
----------------	--------	--

Time Server Payload

The Time Server payload is designated by specifying `com.apple.timeServer` as the `value`.

This payload allows devices to connect to custom time servers.

In addition to the settings common to all payload types, the Time Server payload defines the following keys:

Key	Type	Value
timeServerString	String	The ntp server to connect to. Availability: Available in macOS 10.12.4 and later.
timeZoneString	String	Time zone path location string in /usr/share/zoneinfo, for example, /"America/Denver" or "Zulu". Availability: Available in macOS 10.12.4 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

91

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](#).

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

Key	Type	Value
UserDefinedName	String	Optional. Description of the VPN connection displayed on the device.
VPNType	String	Determines the settings available in the payload for this type of VPN connection. It can have one of the following values: <ul style="list-style-type: none"> • L2TP • PPTP • IPSec (Cisco) • IKE (See IKEv2 Dictionary Keys) • AlwaysOn (See AlwaysOn Dictionary Keys) • VPN (Solution uses a VPN plugin or NetworkExtension so the VPN SubType is required (see below)).
VPNSubType	String	Optional. If VPNType is PPTP, this field is required. If the configuration is targeted at a VPN solution that uses a VPN plugin, then this field contains the bundle identifier of the plugin. Here are some examples: <ul style="list-style-type: none"> • Cisco AnyConnect: com.cisco.anyconnect.applevpn.plugin • Juniper SSL: net.juniper.sslvpn • F5 SSL: com.f5.F5-Edge-Client.vpnplugin

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

91

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](#).

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

Key	Type	Value
<code>UserDefinedName</code>	String	Optional. Description of the VPN connection displayed on the device.
<code>VPNType</code>	String	Determines the settings available in the payload for this type of VPN connection. It can have one of the following values: <ul style="list-style-type: none"> • L2TP • PPTP • IPSec (Qosco) • IKEv2 (IKEv2 Dictionary Keys) • AlwaysOn Dictionary Keys • VPN Resolution uses a VPN plugin or Network Extension so the VPN Sub Type is required (see below).

<code>VPNSubType</code>	String	Optional. If <code>VPNType</code> is L2TP, this field is required. If the configuration is targeted at a VPN solution that uses a VPN plugin, then this field contains the bundle identifier of the plugin. Here are some examples: <ul style="list-style-type: none"> • Cisco AnyConnect: <code>com.cisco.anyconnect.applevpn.plugin</code> • Juniper SSL: <code>com.juniper.sslvpn</code> • F5 SSL: <code>com.f5.F5-Edge-Client.vpnplugin</code> • SonicWALL Mobile Connect: <code>com.sonicwall.SonicWALL-SSLVPN.vpnpp</code> • Aruba VIA: <code>com.arubanetworks.aruba-via.vpnplugin</code> If the configuration is targeted at a VPN solution that uses a Network Extension provider, then this field contains the bundle identifier of the app that contains the provider. Contact the VPN solution vendor for the value of the identifier. If <code>VPNType</code> is IKEv2, then the <code>VPNSubType</code> is optional and is reserved for future use. If it is specified, it must contain the empty string.
<code>ProviderBundleId</code>		

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

92

If `VPNType` is `PPP or IKEv2`, the following keys may be defined in the corresponding `VPNType` dictionaries to configure VPN On Demand:

Key	Type	Value
<code>OnDemandEnable</code>	Boolean	1 if the VPN connection should be brought up on demand, else 0.
<code>OnDemandMatch Array of DomainsAlways</code>	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 <code>OnDemandMatch DomainsOnRetry</code> key. This behavior can be overridden by <code>OnDemandRules</code> .
<code>OnDemandMatch Array of DomainsNever</code>	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 <code>OnDemandRules</code> . In iOS 7 and later, this key is deprecated (but still supported) in favor of <code>EvaluateConnections</code> in the <code>OnDemandRules</code> dictionaries.
<code>OnDemandMatch Array of DomainsOnRetry</code>	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 <code>OnDemandRules</code> . In iOS 7 and later, this key is deprecated (but still supported) in favor of <code>EvaluateConnections</code> in the <code>OnDemandRules</code> dictionaries.
<code>OnDemandRules Array of Dictionaries</code>	Dictionary	Determines when and how an on-demand VPN should be used. See On Demand Rules Dictionary Keys for details.

If `VPNType` is `AlwaysOn`, the following key may be defined:

Key	Type	Value
<code>VendorConfig</code>	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 `VPNTyp` value under which the keys are used.

PPP Dictionary Keys

The following elements are for VPN payloads of type PPP.

Key	Type	Value
<code>AuthName</code>	String	The VPN account user name. Used for L2TP and PPTP.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

93

- SonicWALL Mobile Connect: `com.sonicwall.SonicWALL-SSLVPN.vpnpp`
 - Aruba VIA: `com.arubanetworks.aruba-via.vpnplugin`
- If the configuration is targeted at a VPN solution that uses a Network Extension provider, then this field contains the bundle identifier of the app that contains the provider. Contact the VPN solution vendor for the value of the identifier. If `VPNType` is `AlwaysOn`, then the `VPNSubType` is optional and is reserved for future use. If it is specified, it must contain the empty string.

`ProviderBundleId`

Optional. If the `VPNSubType` contains the bundle identifier of an app that contains multiple VPN providers of the same type (a `pp-provider` key), then this field is used to specify which provider to use for this configuration.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

92

If `VPNType` is `IKEv2`, the following keys may be defined in the corresponding `VPNType` dictionaries to configure VPN On Demand:

Key	Type	Value
<code>OnDemandEnable</code>	Boolean	1 if the VPN connection should be brought up on demand, else 0.
<code>OnDemandMatch Array of DomainsAlways</code>	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 <code>OnDemandMatch DomainsOnRetry</code> key. This behavior can be overridden by <code>OnDemandRules</code> .
<code>OnDemandMatch Array of DomainsNever</code>	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 <code>OnDemandRules</code> . In iOS 7 and later, this key is deprecated (but still supported) in favor of <code>EvaluateConnections</code> in the <code>OnDemandRules</code> dictionaries.
<code>OnDemandMatch Array of DomainsOnRetry</code>	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 <code>OnDemandRules</code> . In iOS 7 and later, this key is deprecated (but still supported) in favor of <code>EvaluateConnections</code> in the <code>OnDemandRules</code> dictionaries.
<code>OnDemandRules Array of Dictionaries</code>	Dictionary	Determines when and how an on-demand VPN should be used. See On Demand Rules Dictionary Keys for details.

If `VPNType` is `AlwaysOn`, the following key may be defined:

Key	Type	Value
<code>VendorConfig</code>	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 `VPNTyp` value under which the keys are used.

PPP Dictionary Keys

The following elements are for VPN payloads of type PPP.

Key	Type	Value
<code>AuthName</code>	String	The VPN account user name. Used for L2TP and PPTP.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

93

Key	Type	Value
<code>AuthPassword</code>	String	Optional. Only visible if TokenCard is a 1 used for L2TP and PPTP.
<code>TokenCard</code>	Boolean	Whether to use a token card such as an RSA SecurID card for connecting. Used for L2TP.
<code>ComRemoteAddress</code>	String	IP address or host name of VPN server. Used for L2TP and PPTP.
<code>AuthEAPPlugIn</code>	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.
<code>AuthProtocol</code>	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.
<code>CCMPPE40Enabled</code>	Boolean	See discussion under <code>CCPENab</code> for PPTP.
<code>CCMPPE128Enabled</code>	Boolean	See discussion under <code>CCPENab</code> for PPTP.
<code>CCPEnabled</code>	Boolean	Enables encryption on the connection. If this key and <code>CCMPPE40</code> are both present, automatic encryption level; if this key and <code>CCMPPE128</code> are both present, maximum encryption level. If no encryption is used, then none of the CCP keys are used for PPTP.

IPv4 Dictionary Keys

The following element is for VPN payloads of type L2TP or PPTP.

Key	Type	Value
<code>OverridePriority</code>	Boolean	Specifies whether to send all traffic through the VPN interface. If 1, all network traffic is sent over VPN. Defaults to 0.

Key	Type	Value
AuthPassword	String	Optional. Only visible if TokenCard is false for L2TP and PPTP.
TokenCard	Boolean	Whether to use a token card such as an RSA SecurID card for connecting. Used for L2TP.
ComRemoteAddress	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 CCPEnabledUsed for PPTP.
CCPMPPE128Enabled	Boolean	See discussion under CCPEnabledUsed for PPTP.
CCPEnabled	Boolean	Enables encryption on the connection. If this key and CCPMPPE40Enabled represents automatic encryption level; if this key and CCPMPPE128Enabled represents maximum encryption level. If no encryption is used, then none of the CCP keys are used for PPTP.

IPv4 Dictionary Keys

The following element is for VPN payloads of type L2TP or PPTP

Key	Type	Value
OverridePriority	String	Specifies whether to send all traffic through the VPN interface. If 1, all network traffic is sent over VPN. Defaults to 0.

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	Either SharedSecret or certificate	Used for L2TP and Cisco IPSec.
XAuthEnabled	Integer	1 if Xauth is on, 0 if it is off. Used for Cisco IPSec.
XAuthName	String	User name for VPN account. Used for Cisco IPSec.
XAuthPassword	String	Required for VPN account user authentication. Used for Cisco IPSec.
LocalIdentifier	String	Present only if AuthenticationMethod is Secret. The name of the group to use. If Hybrid Authentication is used, the string must end with [hybrid] for Cisco IPSec.
LocalIdentifierString	String	Present only if AuthenticationMethod is Secret. The value is KeyUsed for L2TP and Cisco IPSec.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

94

Key	Type	Value
SharedSecret	Data	The shared secret for this VPN account. Only present if AuthenticationMethod is Secret for L2TP and Cisco IPSec.
PayloadCertificateStringUUID	String	The UUID of the certificate to use for the account credentials. Only present if AuthenticationMethod is Certificate for Cisco IPSec.
PromptForPIN	Boolean	Tells whether to prompt for a PIN when connecting. Used for Cisco IPSec.

On Demand Rules Dictionary Keys

The OnDemandRule 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 OnDemandRule 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 OnDemandRule, then for each dictionary in that dictionary's EvaluateConarray, 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 ConnectImmediately or ignore) is used if the dictionary otherwise matches.

Note

For backwards compatibility, VPN On Demand also allows you to specify the All1 action, in which case the domains to match are determined by arrays in the VPN payload itself (OnDemandMatchDomainsAlwaysOnDemandMatchDomains, and OnDemandMatchDomains). 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.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

95

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	Either SharedSecret or certificate	Used for L2TP and Cisco IPSec.
XAuthEnabled	Integer	1 if Xauth is on, 0 if it is off. Used for Cisco IPSec.
XAuthName	String	User name for VPN account. Used for Cisco IPSec.
XAuthPassword	String	Required for VPN account user authentication. Used for Cisco IPSec.
LocalIdentifier	String	Present only if AuthenticationMethod is Secret. The name of the group to use. If Hybrid Authentication is used, the string must end with [hybrid] for Cisco IPSec.
LocalIdentifierString	String	Present only if AuthenticationMethod is Secret. The value is KeyUsed for L2TP and Cisco IPSec.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

94

Key	Type	Value
SharedSecret	Data	The shared secret for this VPN account. Only present if AuthenticationMethod is Secret for L2TP and Cisco IPSec.
PayloadCertificateStringUUID	String	The UUID of the certificate to use for the account credentials. Only present if AuthenticationMethod is Certificate for Cisco IPSec.
PromptForPIN	Boolean	Tells whether to prompt for a PIN when connecting. Used for Cisco IPSec.

On Demand Rules Dictionary Keys

The OnDemandRule 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 OnDemandRule 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 OnDemandRule, then for each dictionary in that dictionary's EvaluateConarray, 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 ConnectImmediately or ignore) is used if the dictionary otherwise matches.

Note

For backwards compatibility, VPN On Demand also allows you to specify the All1 action, in which case the domains to match are determined by arrays in the VPN payload itself (OnDemandMatchDomainsAlwaysOnDemandMatchDomains, and OnDemandMatchDomains). 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.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

95

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).

Note

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 OnDemandRule 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: <ul style="list-style-type: none"> A11 (deprecated). Allow VPN On Demand to connect if triggered. Connect (deprecated). Conditionally initiate a VPN connection on the next network attempt. Disconnect (deprecated). Disconnect on demand as long as this dictionary matches. EvaluateCon (deprecated). An action for each connection attempt.

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).

Note

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 OnDemandDictionaries 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: <ul style="list-style-type: none"> Allow—Deprecated. Allow VPN On Demand to connect if triggered. Connect—Unconditionally initiate a VPN connection on the next network attempt. Disconnect—Shutdown the VPN connection and do not reconnect on demand as long as this dictionary matches. Evaluate—Evaluate the Action parameter 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 Keys in the ActionParameters dictionary .
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.
DNSServerAddress	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, 172.* matches any DNS server in the class A 17 subnet.
InterfaceType	String	An interface type. If specified, this rule matches only if the primary network interface hardware matches the specified type. Supported values are Ethernet, and Cellular.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

96

Key	Type	Value
SSIDMatch	Array of Strings	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:

Key	Type	Value
Domains	Array of Strings	Required. The domains for which this evaluation applies.
DomainAction	String	Required. Defines the VPN behavior for the specified domains. Allowed values are: <ul style="list-style-type: none"> 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 will not trigger a VPN connection nor be accessible through an existing VPN connection.
RequiredDNSServer	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.
RequiredURLStringProbe	String	Optional. An HTTP (HTTP/1.1) URL to probe, using a GET request. If no HTTP response code is received from the server, a VPN connection is established in response.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

97

ActionParameters	Array of Dictionaries	<ul style="list-style-type: none"> Ignore—Leave any existing VPN connection up, but do not reconnect on demand as long as this dictionary matches. <p>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 Keys in the ActionParameters dictionary.</p>
DNSDomainMatch	Array of Strings	<p>A rule matches if any of the domain names in the specified list matches any domain in the device's search domains list.</p> <p>A wildcard '*' prefix is supported. For example, *.example.com matches against either mydomain.example.com or yourdomain.example.com</p>
DNSServerAddress	Array of Strings	<p>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, 172.* matches any DNS server in the class A 17 subnet.</p>
InterfaceType	String	<p>An interface type. If specified, this rule matches only if the primary network interface hardware matches the specified type. Supported values are Ethernet, and Cellular.</p>

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

96

Key	Type	Value
SSIDMatch	Array of Strings	<p>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.</p> <p>Omit this key and the corresponding array to match against any SSID.</p>
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:

Key	Type	Value
Domains	Array of Strings	Required. The domains for which this evaluation applies.
DomainAction	String	Required. Defines the VPN behavior for the specified domains. Allowed values are: <ul style="list-style-type: none"> 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 will not trigger a VPN connection nor be accessible through an existing VPN connection.
RequiredDNSServer	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.
RequiredURLStringProbe	String	Optional. An HTTP (HTTP/1.1) URL to probe, using a GET request. If no HTTP response code is received from the server, a VPN connection is established in response.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

97

IKEv2 Dictionary Keys

If VPNType is IKE, the following keys may be provided in a dictionary:

Key	Type	Value
RemoteAddress	String	Required. IP address or hostname of the VPN server.
LocalIdentifier	String	Required. Identifier of the IKEv2 client in one of the following formats: <ul style="list-style-type: none"> FQDN UserFQDN Address ASN1DN
RemoteIdentifier	String	Required. Remote identifier in one of the following formats: <ul style="list-style-type: none"> FQDN UserFQDN Address ASN1DN
AuthenticationMethod	String	Required. One of the following: <ul style="list-style-type: none"> SharedSecret Certificate

IKEv2 Dictionary Keys

If VPN Tunnel IKE keys following keys may be provided in a dictionary:

Key	Type	Value
RemoteAddress	String	Required. IP address or hostname of the VPN server.
LocalIdentifier	String	Required. Identifier of the IKEv2 client in one of the following formats: <ul style="list-style-type: none"> FQDN User FQDN Address ASN1DN
RemoteIdentifier	String	Required. Remote identifier in one of the following formats: <ul style="list-style-type: none"> FQDN User FQDN Address ASN1DN
AuthenticationMethod	String	Required. One of the following: <ul style="list-style-type: none"> SharedSecret Certificate None To enable EAP-only authentication, the authentication method should be set to None and the ExtendedAuthKey is enabled should be set to 1. If this key is set to None and the ExtendedAuthKey is not set, the authentication configuration defaults to SharedSecret.
PayloadCertificateUUID	String	Optional. The UUID of the identity certificate used as the account credential. If the value of AuthenticationMethod Certificate, this certificate is sent out for IKEv2 machine authentication. If extended authentication (EAP) is used, it is sent out for EAP-TLS authentication.
CertificateTypeString	String	Optional. This key specifies the type of PayloadCertificate used for IKEv2 machine authentication. Its value is one of the following: <ul style="list-style-type: none"> RSA (Default) ECDSA 256 ECDSA 384 ECDSA 521 If this key is included, the ServerCertificateIssuer is required.
SharedSecret	String	Optional. If AuthenticationMethod is set to SharedSecret, this value is used for IKE authentication.
ExtendedAuthEnabled	Integer	Optional. Set to 1 to enable EAP-only authentication (see AuthenticationMethod). Defaults to 0.
AuthName	String	Optional. Username used for authentication.
AuthPassword	String	Optional. Password used for authentication.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

98

Key	Type	Value
DeadPeerDetectionRate	String	Optional. One of the following: <ul style="list-style-type: none"> None (Disable) LowKeepalive every 30 minutes MediumKeepalive every 10 minutes HighKeepalive every 1 minute Defaults to Medium.
ServerCertificateIssuerCommonName	String	Optional. Common Name of the server certificate issuer. If set, this field will cause IKE to send a certificate request based on this certificate issuer to the server. This key is required if both the CertificateKey is specified and the ExtendedAuthKey is set to 1.
ServerCertificateCommonName	String	Optional. Common Name of the server certificate. This name is used to validate the certificate sent by the IKE server. If not set, the RemoteIdentifier will be used to validate the certificate.
TLSMinimumVersion	String	Optional. The minimum TLS version to be used with EAP-TLS authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default minimum is 1.0.
TLSMaximumVersion	String	Optional. The maximum TLS version to be used with EAP-TLS authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default maximum is 1.2.
NATKeepaliveOffloadEnable	String	Optional. Set to 1 to enable or 0 to disable NAT Keepalive offload for Always On VPN IKEv2 connections. Keepalive packets are sent by the device to maintain NAT mappings for IKEv2 connections that have a NAT on the path. Keepalive packets are sent at regular interval when the device is awake. If NATKeepaliveOffload is set to 1, Keepalive packets will be offloaded to hardware while the device is asleep. NAT Keepalive offload has an impact on the battery life since extra workload is added during sleep. The default interval for the Keepalive offload packets is 20 seconds over WiFi and 110 seconds over Cellular interface. The default NAT Keepalive works well on networks with small NAT mapping timeouts but imposes a potential battery impact. If a network is known to have larger NAT mapping timeouts, larger Keepalive intervals may be safely used to minimize battery impact. The Keepalive interval can be modified by setting the NATKeepaliveKey. Default value for NATKeepaliveOffload is 1.
NATKeepaliveInterval	String	Optional. NAT Keepalive interval for Always On VPN IKEv2 connections. This value controls the interval over which Keepalive offload packets are sent by the device. The minimum value is 20 seconds. If no key is specified, the default is 20 seconds over WiFi and 110 seconds over a Cellular interface.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

98

Key	Type	Value
DeadPeerDetectionRate	String	Optional. One of the following: <ul style="list-style-type: none"> None (Disable) LowKeepalive every 30 minutes MediumKeepalive every 10 minutes HighKeepalive every 1 minute Defaults to Medium.
ServerCertificateIssuerCommonName	String	Optional. Common Name of the server certificate issuer. If set, this field will cause IKE to send a certificate request based on this certificate issuer to the server. This key is required if both the CertificateKey is specified and the ExtendedAuthKey is set to 1.
ServerCertificateCommonName	String	Optional. Common Name of the server certificate. This name is used to validate the certificate sent by the IKE server. If not set, the RemoteIdentifier will be used to validate the certificate.
TLSMinimumVersion	String	Optional. The minimum TLS version to be used with EAP-TLS authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default minimum is 1.0.
TLSMaximumVersion	String	Optional. The maximum TLS version to be used with EAP-TLS authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default maximum is 1.2.
NATKeepaliveOffloadEnable	String	Optional. Set to 1 to enable or 0 to disable NAT Keepalive offload for Always On VPN IKEv2 connections. Keepalive packets are sent by the device to maintain NAT mappings for IKEv2 connections that have a NAT on the path. Keepalive packets are sent at regular interval when the device is awake. If NATKeepaliveOffload is set to 1, Keepalive packets will be offloaded to hardware while the device is asleep. NAT Keepalive offload has an impact on the battery life since extra workload is added during sleep. The default interval for the Keepalive offload packets is 20 seconds over WiFi and 110 seconds over Cellular interface. The default NAT Keepalive works well on networks with small NAT mapping timeouts but imposes a potential battery impact. If a network is known to have larger NAT mapping timeouts, larger Keepalive intervals may be safely used to minimize battery impact. The Keepalive interval can be modified by setting the NATKeepaliveKey. Default value for NATKeepaliveOffload is 1.
NATKeepaliveInterval	String	Optional. NAT Keepalive interval for Always On VPN IKEv2 connections. This value controls the interval over which Keepalive offload packets are sent by the device. The minimum value is 20 seconds. If no key is specified, the default is 20 seconds over WiFi and 110 seconds over a Cellular interface.
UseConfigurationAttributeInternalIPSubnet	Integer	Optional. If set to 1, negotiations should use IKEv2 Configuration Attribute INTERNAL_IP4_SUBNET and INTERNAL_IP6_SUBNET. Defaults to 0.
DisableMOBIKE	Integer	Optional. If set to 1, disables MOBIKE. Defaults to 0.
DisableRedirect	Integer	Optional. If set to 1, disables IKEv2 redirect. If not set, the IKEv2 connection would be redirected if a redirect request is received from the server. Defaults to 0.
EnablePFS	Integer	Optional. Set to 1 to enable Perfect Forward Secrecy (PFS) for IKEv2 Connections. Default is 0.
EnableCertificateRevocationCheck	Integer	Optional. Set to 1 to enable a certificate revocation check for IKEv2 connections. This is a best-effort revocation check; server

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

99

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

99

Key	Type	Value
UseConfigurationIntegerAttributeInternalIPSubnet	Integer	Optional. If set to 1, negotiations should use IKEv2 Configuration Attribute INTERNAL_IP4_SUBNET and INTERNAL_IP6_SUBNET. Defaults to 0.
Availability:		Available in iOS 9.0 and later.
DisableMOBIKE	Integer	Optional. If set to 1, disables MOBIKE. Defaults to 0.
Availability:		Available in iOS 9.0 and later.
DisableRedirect	Integer	Optional. If set to 1, disables IKEv2 redirect. If not set, the IKEv2 connection would be redirected if a redirect request is received from the server. Defaults to 0.
Availability:		Available in iOS 9.0 and later.
EnablePFS	Integer	Optional. Set to 1 to enable Perfect Forward Secrecy (PFS) for IKEv2 Connections. Default is 0.
Availability:		Available in iOS 9.0 and later.
EnableCertificateRevocationCheck	Integer	Optional. Set to 1 to enable a certificate revocation check for IKEv2 connections. This is a best-effort revocation check; server response timeouts will not cause it to fail.
Availability:		Available in iOS 9.0 and later.
IKESecurityAssociationParameters	Dictionary	Optional. See table below. Applies to child Security Association unless ChildSecurityAssociationParameters is specified.
ChildSecurityDictionary	Dictionary	Optional. See table below.
AssociationParameters	Dictionary	Optional. See table below.

The IKE Security Association and Child Security Association parameters contain the following keys:

Key	Type	Value
EncryptionAlgorithm	String	Optional. One of: <ul style="list-style-type: none"> DES 3DES AES-128 AES-256 (Default) AES-128-GCM (16-octet ICV) AES-256-GCM (16-octet ICV)
IntegrityAlgorithm	String	Optional. One of: <ul style="list-style-type: none"> SHA-96 SHA-1-160 SHA-2-256 (Default) SHA-2-384 SHA-2-512
DiffieHellmanGroup	Integer	Optional. One of: 1, 2, 5, 14 (Default), 15, 16, 17, 18, 19, 20, or 21.
LifeTimeInMinutes	Integer	Optional. SA lifetime (rekey interval) in minutes. Valid values are 10 through 1440. Defaults to 1440 minutes.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

100

DNS Dictionary Keys

If VPN Tunnel is enabled the following DNS keys may be provided:

Key	Type	Value
ServerAddresses	Array of Strings	Required. An array of DNS server IP address strings. These IP addresses can be a mixture of IPv4 and IPv6 addresses.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
SearchDomains	Array of Strings	Optional. A list of domain strings used to fully qualify single-label host names.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
DomainName	String	Optional. The primary domain of the tunnel.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
SupplementalDomains	Array of Strings	Optional. A list of domain strings used to determine which DNS queries will use the DNS resolver settings contained in ServerAddress . This key is used to create a split DNS configuration where only hosts in certain domains are resolved using the tunnel's DNS resolver. Hosts not in one of the domains in this list are resolved using the system's default resolver.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
SupplementalDomainsNoSearch	Boolean	Optional. Whether (0) or not (1) the domains in the SupplementalDomains should be appended to the resolver's list of search domains. Default is 0.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.

Proxies Dictionary Keys

The Proxies dictionary may contain the following keys:

Key	Type	Value
ProxyAutoConfigEnabled	Integer	Optional. Set to 1 to enable automatic proxy configuration. Defaults to 0.

IKE Security Association Parameters	Dictionary	Optional. See table below. Applies to child Security Association unless ChildSecurityAssociationParameters is specified.
ChildSecurityDictionary	Dictionary	Optional. See table below.

The IKE Security Association and Child Security Association parameters may contain the following keys:

Key	Type	Value
EncryptionAlgorithm	String	Optional. One of: <ul style="list-style-type: none"> DES 3DES AES-128 AES-256 (Default) AES-128-GCM (16-octet ICV) AES-256-GCM (16-octet ICV)
IntegrityAlgorithm	String	Optional. One of: <ul style="list-style-type: none"> SHA-96 SHA-1-160 SHA-2-256 (Default) SHA-2-384 SHA-2-512
DiffieHellmanGroup	Integer	Optional. One of: 1, 2, 5, 14 (Default), 15, 16, 17, 18, 19, 20, or 21.
LifeTimeInMinutes	Integer	Optional. SA lifetime (rekey interval) in minutes. Valid values are 10 through 1440. Defaults to 1440 minutes.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

100

DNS Dictionary Keys

If VPN Tunnel is enabled the following DNS keys may be provided:

Key	Type	Value
ServerAddresses	Array of Strings	Required. An array of DNS server IP address strings. These IP addresses can be a mixture of IPv4 and IPv6 addresses.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
SearchDomains	Array of Strings	Optional. A list of domain strings used to fully qualify single-label host names.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
DomainName	String	Optional. The primary domain of the tunnel.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
SupplementalDomains	Array of Strings	Optional. A list of domain strings used to determine which DNS queries will use the DNS resolver settings contained in ServerAddress . This key is used to create a split DNS configuration where only hosts in certain domains are resolved using the tunnel's DNS resolver. Hosts not in one of the domains in this list are resolved using the system's default resolver.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.
SupplementalDomainsNoSearch	Boolean	Optional. Whether (0) or not (1) the domains in the SupplementalDomains should be appended to the resolver's list of search domains. Default is 0.
Availability:		Available in iOS 10.0 and later and macOS 10.12 and later.

Proxies Dictionary Keys

The Proxies dictionary may contain the following keys:

Key	Type	Value
ProxyAutoConfigEnabled	Integer	Optional. Set to 1 to enable automatic proxy configuration. Defaults to 0.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

101

Key	Type	Value
ProxyAutoConfigString	String	Optional. URL to the location of the proxy auto-configuration file. Used only when ProxyAutoConfigEnabled is 1.
SupplementalDomains	Array of Strings	Optional. If set, then only connections to hosts within one or more of the specified domains will use the proxy settings

The Proxy Auto Configuration dictionary may also contain the following keys:

Key	Type	Value
HTTPEnable	Integer	Optional. Set to 1 to enable proxy for HTTP traffic. Defaults to 0.
HTTPProxy	String	Optional. The host name of the HTTP proxy.
HTTPPort	Integer	Optional. The port number of the HTTP proxy. This field is required if HTTPEnable is 1.

Defaults to 0.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

101

Key	Type	Value
ProxyAutoConfigUrl	String	Optional. URL to the location of the proxy auto-configuration file. Used only when ProxyAutoConfig is enabled.
SupplementalMatchStrings	Array of Strings	Optional. If set, then only connections to hosts within one or more of the specified domains will use the proxy settings

If ProxyAutoConfig is 0, the dictionary may also contain the following keys:

Key	Type	Value
HTTPEnable	Integer	Optional. Set to 1 to enable proxy for HTTP traffic. Defaults to 0.
HTTPProxy	String	Optional. The host name of the HTTP proxy.
HTTPPort	Integer	Optional. The port number of the HTTP proxy. This field is required if HTTPProxy is specified.
HTTPProxyUsername	String	Optional. The username used for authentication.
HTTPProxyPassword	String	Optional. The password used for authentication.
HTTPSEnable	Integer	Optional. Set to 1 to enable proxy for HTTPS traffic. Defaults to 0.
HTTPSPort	String	Optional. The host name of the HTTPS proxy.
HTTPSPort	Integer	Optional. The port number of the HTTPS proxy. This field is required if HTTPSPort is specified.

AlwaysOn Dictionary Keys

If VPN is 1, the dictionary may contain the following keys:

Key	Type	Value
UIToggleEnabled	Integer	Optional. If set to 1, allows the user to disable this VPN configuration. Defaults to 0.
TunnelConfiguration	Array of Dictionaries	Required. See below.
ServiceExceptions	Array of Dictionaries	Optional. See below.
AllowCaptiveWebSheet	Integer	Optional. Set to 1 to allow traffic from Captive Web Sheet outside the VPN tunnel. Defaults to 0.
AllowAllCaptiveNetworkPlugins	Integer	Optional. Set to 1 to allow traffic from all Captive Networking apps outside the VPN tunnel to perform Captive network handling. Defaults to 0.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

102

Key	Type	Value
AllowedCaptiveNetworkPlugins	Array of Dictionaries	Optional. Array of Captive Networking apps whose traffic will be allowed outside the VPN tunnel to perform Captive network handling. Used only when AllowAllCaptiveNetworkPlugins is 1. Each dictionary in the AllowedCaptiveNetworkPlugins array contains a <code>BundleIdentifier</code> of type string, the value of which is the app's bundle identifier. Captive Networking apps may require additional entitlements to operate in a captive environment.

Each dictionary in a TunnelConfig array may contain the following keys:

Key	Type	Value
Protocol	String	Must be IKEv2.
Interface	Array of Strings	Optional. Specify the interfaces to which this configuration applies. Valid values are Cellular and WiFi. Defaults to Cellular, WiFi.

In addition, all keys defined for the IKEv2 dictionary, such as RemoteAddress, LocalInterface, and `Identifier`, are present in a TunnelConfig dictionary.

Each dictionary in a ServiceExceptions array may contain the following keys:

Key	Type	Value
ServiceName	String	Required. The name of a system service which is exempt from Always On VPN. Must be one of: <ul style="list-style-type: none"> • Voicemail • AirPrint • Cellular (Available in iOS 11.3 and later.)
Action	String	Required. One of the following: <ul style="list-style-type: none"> • Allow • Drop

HTTPProxy	String	Optional. The host name of the HTTP proxy.
HTTPProxyUsername	String	Optional. The username used for authentication.
HTTPProxyPassword	String	Optional. The password used for authentication.
HTTPSEnable	Integer	Optional. Set to 1 to enable proxy for HTTPS traffic. Defaults to 0.
HTTPSPort	String	Optional. The host name of the HTTPS proxy.
HTTPSPort	Integer	Optional. The port number of the HTTPS proxy. This field is required if HTTPSPort is specified.

AlwaysOn Dictionary Keys

If VPN is 1, the dictionary may contain the following keys:

Key	Type	Value
UIToggleEnabled	Integer	Optional. If set to 1, allows the user to disable this VPN configuration. Defaults to 0.
TunnelConfiguration	Array of Dictionaries	Required. See below.
ServiceExceptions	Array of Dictionaries	Optional. See below.
AllowCaptiveWebSheet	Integer	Optional. Set to 1 to allow traffic from Captive Web Sheet outside the VPN tunnel. Defaults to 0.
AllowAllCaptiveNetworkPlugins	Integer	Optional. Set to 1 to allow traffic from all Captive Networking apps outside the VPN tunnel to perform Captive network handling. Defaults to 0.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

102

Key	Type	Value
AllowedCaptiveNetworkPlugins	Array of Dictionaries	Optional. Array of Captive Networking apps whose traffic will be allowed outside the VPN tunnel to perform Captive network handling. Used only when AllowAllCaptiveNetworkPlugins is 1. Each dictionary in the AllowedCaptiveNetworkPlugins array contains a <code>BundleIdentifier</code> of type string, the value of which is the app's bundle identifier. Captive Networking apps may require additional entitlements to operate in a captive environment.

Each dictionary in a TunnelConfig array may contain the following keys:

Key	Type	Value
Protocol	String	Must be IKEv2.
Interface	Array of Strings	Optional. Specify the interfaces to which this configuration applies. Valid values are Cellular and WiFi. Defaults to Cellular, WiFi.

In addition, all keys defined for the IKEv2 dictionary, such as RemoteAddress, LocalInterface, and `Identifier`, are present in a TunnelConfig dictionary.

Each dictionary in a ServiceExceptions array may contain the following keys:

Key	Type	Value
ServiceName	String	Required. The name of a system service which is exempt from Always On VPN. Must be one of: <ul style="list-style-type: none"> • Voicemail • AirPrint • Cellular (Available in iOS 11.3 and later.)
Action	String	Required. One of the following: <ul style="list-style-type: none"> • Allow • Drop

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

103

Per-App VPN Payload

The Per-App VPN payload is used for configuring add-on VPN software, and it works only on VPN services of type 'VPN'. It should not be confused with the standard VPN payload, described in [VPN Payload](#).

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

The VPN payload is designated by specifying `com.apple.vpn.managed.Bpple`. The Per-App VPN payload supports all of the keys described in [VPN Payload](#) plus the following additional keys:

Key	Type	Value
VPNUUID	String	A globally-unique identifier for this VPN configuration. This

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

103

Per-App VPN Payload

The Per-App VPN payload is used for configuring add-on VPN software, and it works only on VPN services of type 'VPN'. It should not be confused with the standard VPN payload, described in [VPN Payload](#).

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

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

Key	Type	Value
<code>VPNUUID</code>	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 .
<code>SafariDomains</code>	Array	This optional key is a special case of App-to-Per App VPN Mapping. It sets up the app mapping for Safari (Webkit) with a specific identifier and a designated requirement. The array contains strings, each of which is a domain that should trigger this VPN connection in Safari. The rule matching behavior is as follows: <ul style="list-style-type: none"> 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 "example.matches" www.example.com, but not "foo.bade.example.com". Domain strings with only one label must match the entire host string. For example, a domain of "com.matches" "com" www.example.com".
<code>OnDemandMatchAppPayload</code>	Boolean	Indicates whether the Per-App VPN connection starts automatically when apps linked to this Per-App VPN service initiate network communication. If a 1 is set, 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 <code>OnDemandEnabled</code> key is used to determine the status of Per-App VPN On Demand. Optional. Either <code>packet-tunnel</code> or <code>ip</code> . The default is <code>ip</code> . If the value of this key is <code>packet-tunnel</code> , then the VPN service will tunnel traffic at the application layer. If the value of this key is <code>ip</code> , then the VPN service will tunnel traffic at the IP layer.
<code>ProviderType</code>	String	

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

104

App-to-Per-App VPN Mapping

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

This payload is supported only in macOS 10.9 and later. It is not supported in iOS.

Key	Type	Value
<code>AppPlayerVPNMapping</code>	Array of Dictionaries	An array of mapping dictionaries.
Each dictionary in the array must contain the following keys:		
<code>Identifier</code>	String	Required. The app's bundle ID.
<code>VPNUUID</code>	String	Required. The VPNUUID of the Per-App VPN defined in a Per-App VPN payload.
<code>DesignatedRequirement</code>	String	Required. The code signature designated requirement of the app that will use the per-app VPN.
<code>SigningIdentifier</code>	String	Required. The code signature signing identifier of the app that will use the per-app VPN.

Web Clip Payload

The Web Clip payload is designated by specifying `com.apple.webClip` as the `fnayield` 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
<code>URL</code>	String	The URL that the Web Clip should open when clicked. The URL must begin with HTTP or HTTPS or it won't work.
<code>Label</code>	String	The name of the Web Clip as displayed on the Home screen.
<code>Icon</code>	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.
<code>IsRemovable</code>	Boolean	Optional. If a 1 is set, the web clip is not removable. Defaults to true. Not available in macOS.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

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](#).

This optional key is a special case of App-to-Per App VPN Mapping. It sets up the app mapping for Safari (Webkit) with a specific identifier and a designated requirement. The 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 "example.matches" www.example.com, but not "foo.bade.example.com".
- Domain strings with only one label must match the entire host string. For example, a domain of "com.matches" "com" www.example.com".

`OnDemandMatchAppPayload` indicates whether the Per-App VPN connection starts automatically when apps linked to this Per-App VPN service initiate network communication.

If a 1 is set, 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.

Optional. Either `packet-tunnel` or `ip`. The default is `ip`. If the value of this key is `packet-tunnel`, then the VPN service will tunnel traffic at the application layer. If the value of this key is `ip`, then the VPN service will tunnel traffic at the IP layer.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

104

App-to-Per-App VPN Mapping

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

This payload is supported only in macOS 10.9 and later. It is not supported in iOS.

Key	Type	Value
<code>AppPlayerVPNMapping</code>	Array of Dictionaries	An array of mapping dictionaries.
Each dictionary in the array must contain the following keys:		
<code>Identifier</code>	String	Required. The app's bundle ID.
<code>VPNUUID</code>	String	Required. The VPNUUID of the Per-App VPN defined in a Per-App VPN payload.
<code>DesignatedRequirement</code>	String	Required. The code signature designated requirement of the app that will use the per-app VPN.
<code>SigningIdentifier</code>	String	Required. The code signature signing identifier of the app that will use the per-app VPN.

Web Clip Payload

The Web Clip payload is designated by specifying `com.apple.webClip` as the `fnayield` 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
<code>URL</code>	String	The URL that the Web Clip should open when clicked. The URL must begin with HTTP or HTTPS or it won't work.
<code>Label</code>	String	The name of the Web Clip as displayed on the Home screen.
<code>Icon</code>	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.
<code>IsRemovable</code>	Boolean	Optional. If a 1 is set, the web clip is not removable. Defaults to true. Not available in macOS.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

105

Web Content Filter Payload

The Web Content Filter payload allows you to whitelist and blacklist specific web URLs. This payload is supported only on supervised devices.

Web content filtering is designated by specifying `com.apple.webcontentfilter` as the `fnayield` value and adding a `filter` string with one of these values:

- `BuildDefault`

available in macOS.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

105

Web Content Filter Payload

The Web Content Filter payload allows you to whitelist and blacklist specific web URLs. This payload is supported only on supervised devices.

Web content filtering is designated by specifying `com.apple.webcontentfilter` as the `Payload` and adding a `Filter` `Type` with one of these values:

- `Build` (Default)
- `Plugin`

On macOS, `Filter` `Type` `Plugin`

If `Filter` `Type` `Plugin`, this payload defines the following keys in addition to the settings common to all payloads:

Key	Type	Value
<code>AutoFilterEnabled</code>	Boolean	Optional. If true, automatic filtering is enabled. This function evaluates each web page as it is loaded and attempts to identify and block content not suitable for children. The search algorithm is complex and may vary from release to release, but it is basically looking for adult language, i.e. swearing and sexually explicit language. The default value is <code>false</code> .
<code>PermittedURLs</code>	Array of Strings	Optional. Used only when <code>AutoFilterEnabled</code> is true. Otherwise, this field is ignored. Each entry contains a URL that is accessible whether the automatic filter allows access or not.
<code>WhitelistedBookmarks</code>	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. The number of these URLs should be limited to about 500.
<code>BlacklistedURLs</code>	Array of Strings	Optional. Access to the specified URLs is blocked. The number of these URLs should be limited to about 500.

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

Key	Type	Value
<code>URL</code>	String	URL of the whitelisted bookmark.
<code>BookmarkFolder</code>	String	Optional. The folder into which the bookmark should be added in Safari—/Interesting Topic Page/Video Log. If absent, the bookmark is added to the default bookmarks directory.
<code>Title</code>	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

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

106

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 by using string-based root matching. A URL matches a whitelist, blacklist, or permitted list pattern if the exact characters of the pattern appear as the root of the URL. For example, if `test.com` is blacklisted, then `test.com` and `test.com/www` will be blocked. Matching does not discard subdomain prefixes, so if `test.com` is blacklisted, `test` is not blocked. Also, no attempt is made to match aliases (IP address versus DNS names, for example) or to handle requests with explicit port numbers.

If a profile does not contain an array for `PermitteableURLList` `elistedB` `q` `that` `profile` is skipped when evaluating the missing array or arrays. As an exception, if a payload contains an `AutoFilter` `key` but does not contain a `PermitteableURLList`, 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.

Apple domains cannot be blacklisted.

If `Filter` `Type` `Plugin`, this payload defines the following keys in addition to the settings common to all payloads:

Key	Type	Value
<code>UserDefinedName</code>	String	A string which will be displayed for this filtering configuration.
<code>PluginBundleID</code>	String	The Bundle ID of the plugin that provides filtering service.
<code>ServerAddress</code>	String	Optional. Server address (may be IP address, hostname, or URL).
<code>UserName</code>	String	Optional. A username for the service.
<code>Password</code>	String	Optional. A password for the service.
<code>PayloadCertificate</code>	String	Optional. UUID pointing to an identity certificate payload. This identity will be used to authenticate the user to the service.
<code>Organization</code>	String	Optional. An Organization string that will be passed to the 3rd-party plugin.
<code>VendorConfig</code>	Dictionary	Optional. Custom dictionary needed by the filtering service plugin.
<code>FilterBrowsers</code>	Integer	Optional. If set to 1, filter WebKit traffic. Defaults to 0.
<code>FilterSockets</code>	Integer	Optional. If set to 1, filter socket traffic. Defaults to 0.

At least one of `FilterBrowsers` or `FilterSockets` must be set for the filter to have any effect.

Plugin

On macOS, `Filter` `Type` `Plugin`

If `Filter` `Type` `Plugin`, this payload defines the following keys in addition to the settings common to all payloads:

Key	Type	Value
<code>AutoFilterEnabled</code>	Boolean	Optional. If true, automatic filtering is enabled. This function evaluates each web page as it is loaded and attempts to identify and block content not suitable for children. The search algorithm is complex and may vary from release to release, but it is basically looking for adult language, i.e. swearing and sexually explicit language. The default value is <code>false</code> .
<code>PermittedURLs</code>	Array of Strings	Optional. Used only when <code>AutoFilterEnabled</code> is true. Otherwise, this field is ignored. Each entry contains a URL that is accessible whether the automatic filter allows access or not.
<code>WhitelistedBookmarks</code>	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. The number of these URLs should be limited to about 500.
<code>BlacklistedURLs</code>	Array of Strings	Optional. Access to the specified URLs is blocked. The number of these URLs should be limited to about 500.

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

Key	Type	Value
<code>URL</code>	String	URL of the whitelisted bookmark.
<code>BookmarkFolder</code>	String	Optional. The folder into which the bookmark should be added in Safari—/Interesting Topic Page/Video Log. If absent, the bookmark is added to the default bookmarks directory.
<code>Title</code>	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

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

106

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 by using string-based root matching. A URL matches a whitelist, blacklist, or permitted list pattern if the exact characters of the pattern appear as the root of the URL. For example, if `test.com` is blacklisted, then `test.com` and `test.com/www` will be blocked. Matching does not discard subdomain prefixes, so if `test.com` is blacklisted, `test` is not blocked. Also, no attempt is made to match aliases (IP address versus DNS names, for example) or to handle requests with explicit port numbers.

If a profile does not contain an array for `PermitteableURLList` `elistedB` `q` `that` `profile` is skipped when evaluating the missing array or arrays. As an exception, if a payload contains an `AutoFilter` `key` but does not contain a `PermitteableURLList`, 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.

Apple domains cannot be blacklisted.

If `Filter` `Type` `Plugin`, this payload defines the following keys in addition to the settings common to all payloads:

Key	Type	Value
<code>UserDefinedName</code>	String	A string which will be displayed for this filtering configuration.
<code>PluginBundleID</code>	String	The Bundle ID of the plugin that provides filtering service.
<code>ServerAddress</code>	String	Optional. Server address (may be IP address, hostname, or URL).
<code>UserName</code>	String	Optional. A username for the service.
<code>Password</code>	String	Optional. A password for the service.
<code>PayloadCertificate</code>	String	Optional. UUID pointing to an identity certificate payload. This identity will be used to authenticate the user to the service.
<code>Organization</code>	String	Optional. An Organization string that will be passed to the 3rd-party plugin.
<code>VendorConfig</code>	Dictionary	Optional. Custom dictionary needed by the filtering service plugin.
<code>FilterBrowsers</code>	Integer	Optional. If set to 1, filter WebKit traffic. Defaults to 0.
<code>FilterSockets</code>	Integer	Optional. If set to 1, filter socket traffic. Defaults to 0.

At least one of `FilterBrowsers` or `FilterSockets` must be set for the filter to have any effect.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

107

Wi-Fi Payload

The Wi-Fi payload is designated by specifying `com.apple.wifi` as the `Payload`.

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

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

107

Wi-Fi Payload

The Wi-Fi payload is designated by specifying `com.apple.wifi` as the `payload` value.

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

Key	Type	Value
<code>SSID_STR</code>	String	SSID of the Wi-Fi network to be used. In iOS 7.0 and later, this is optional if a <code>domainName</code> value is provided.
<code>HIDDEN_NETWORK</code>	Boolean	Besides SSID, the device uses information such as broadcast type and encryption type to differentiate a network. By default (if a 1 is assumed that all configured networks are open or broadcast. To specify a hidden network, must be true. Optional. Default to false if the network is auto-joined. If a 1 the user has to tap the network name to join.
<code>AutoJoin</code>	Boolean	Optional. Default to false if the network is auto-joined. If a 1 the user has to tap the network name to join.
<code>Availability</code>		Available in iOS 5.0 and later and in all versions of macOS.
<code>SetupModes</code>	Array of Strings	Optional. On macOS 802.1x networks, this is an array of strings that contain the type of connection mode which will be attached. Allowed values are <code>System</code> and/or <code>LogInWindow</code>
<code>EncryptionType</code>	String	The possible values are <code>WEPA</code> , <code>WPA2</code> , <code>Both</code> , and <code>NoWPA</code> . <code>WPA</code> specifies WPA only; WPA2 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 <code>Any</code> .
<code>Availability</code>		Key available in iOS 4.0 and later and in all versions of macOS. The <code>NoWPA</code> value is available in iOS 5.0 and later and the <code>WPA</code> value is available in iOS 8.0 and later.
<code>IsHotspot</code>	Boolean	Optional. Default to false if the network is treated as a hotspot.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>DomainName</code>	String	Optional. Domain Name used for Wi-Fi Hotspot 2.0 negotiation. This field can be provided instead of <code>SSID_STR</code> .
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>ServiceProviderEnabled</code>	Boolean	Optional. If true, allows connection to roaming service providers. Defaults to false.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>RoamingConsortium</code>	Array of Strings	Optional. Array of Roaming Consortium Organization Identifiers used for Wi-Fi Hotspot 2.0 negotiation.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

108

Key	Type	Value
<code>NAIRrealmNames</code>	Array of Strings	Optional. Array of strings. List of Network Access Identifier Realm names used for Wi-Fi Hotspot 2.0 negotiation.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>MCCAndMNCs</code>	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.
<code>Availability</code>		Available in iOS 7.0 and later. This feature is not supported in macOS.
<code>DisplayedOperator</code>	String	The operator name to display when connected to this network. Used only with Wi-Fi Hotspot 2.0 access points.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>ProxyType</code>	String	Optional. Valid values are <code>NoNone</code> , <code>Auto</code> , and <code>Manual</code> .
<code>Availability</code>		Available in iOS 5.0 and later and on all versions of macOS.
<code>CaptiveBypass</code>	Boolean	Optional. If set to true, Captive Network detection will be bypassed when the device connects to the network. Defaults to false.
<code>Availability</code>		Available in iOS 10.0 and later.
<code>QoSmarkingPolicy</code>	Dictionary	Optional. When this dictionary is not present for a Wi-Fi network, all apps are whitelisted to use L2 and L3 marking when the Wi-Fi network supports Cisco QoS fast lane. When present in the Wi-Fi payload, the <code>QoSmarkingPolicy</code> dictionary should contain the list of apps that are allowed to benefit from L2 and L3 marking. For dictionary keys, see the table below.
<code>Availability</code>		Available in iOS 10.0 and later and in macOS 10.13 and later.

Key	Type	Value
<code>SSID_STR</code>	String	SSID of the Wi-Fi network to be used. In iOS 7.0 and later, this is optional if a <code>domainName</code> value is provided.
<code>HIDDEN_NETWORK</code>	Boolean	Besides SSID, the device uses information such as broadcast type and encryption type to differentiate a network. By default (if a 1 is assumed that all configured networks are open or broadcast. To specify a hidden network, must be true. Optional. Default to false if the network is auto-joined. If a 1 the user has to tap the network name to join.
<code>AutoJoin</code>	Boolean	Optional. Default to false if the network is auto-joined. If a 1 the user has to tap the network name to join.
<code>Availability</code>		Available in iOS 5.0 and later and in all versions of macOS.
<code>SetupModes</code>	Array of Strings	Optional. On macOS 802.1x networks, this is an array of strings that contain the type of connection mode which will be attached. Allowed values are <code>System</code> and/or <code>LogInWindow</code>
<code>EncryptionType</code>	String	The possible values are <code>WEPA</code> , <code>WPA2</code> , <code>Both</code> , and <code>NoWPA</code> . <code>WPA</code> specifies WPA only; WPA2 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 <code>Any</code> .
<code>Availability</code>		Key available in iOS 4.0 and later and in all versions of macOS. The <code>NoWPA</code> value is available in iOS 5.0 and later and the <code>WPA</code> value is available in iOS 8.0 and later.
<code>IsHotspot</code>	Boolean	Optional. Default to false if the network is treated as a hotspot.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>DomainName</code>	String	Optional. Domain Name used for Wi-Fi Hotspot 2.0 negotiation. This field can be provided instead of <code>SSID_STR</code> .
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>ServiceProviderEnabled</code>	Boolean	Optional. If true, allows connection to roaming service providers. Defaults to false.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>RoamingConsortium</code>	Array of Strings	Optional. Array of Roaming Consortium Organization Identifiers used for Wi-Fi Hotspot 2.0 negotiation.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

108

Key	Type	Value
<code>NAIRrealmNames</code>	Array of Strings	Optional. Array of strings. List of Network Access Identifier Realm names used for Wi-Fi Hotspot 2.0 negotiation.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>MCCAndMNCs</code>	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.
<code>Availability</code>		Available in iOS 7.0 and later. This feature is not supported in macOS.
<code>DisplayedOperator</code>	String	The operator name to display when connected to this network. Used only with Wi-Fi Hotspot 2.0 access points.
<code>Availability</code>		Available in iOS 7.0 and later and in macOS 10.9 and later.
<code>ProxyType</code>	String	Optional. Valid values are <code>NoNone</code> , <code>Auto</code> , and <code>Manual</code> .
<code>Availability</code>		Available in iOS 5.0 and later and on all versions of macOS.
<code>CaptiveBypass</code>	Boolean	Optional. If set to true, Captive Network detection will be bypassed when the device connects to the network. Defaults to false.
<code>Availability</code>		Available in iOS 10.0 and later.
<code>QoSmarkingPolicy</code>	Dictionary	Optional. When this dictionary is not present for a Wi-Fi network, all apps are whitelisted to use L2 and L3 marking when the Wi-Fi network supports Cisco QoS fast lane. When present in the Wi-Fi payload, the <code>QoSmarkingPolicy</code> dictionary should contain the list of apps that are allowed to benefit from L2 and L3 marking. For dictionary keys, see the table below.
<code>Availability</code>		Available in iOS 10.0 and later and in macOS 10.13 and later.

The `QoSmarkingPolicy` dictionary contains these keys:

Key	Type	Value
<code>QoSmarkingWhitelistedAppIdentifiers</code>	Array of Strings	Optional. Array of app bundle identifiers that will be whitelisted for L2 and L3 marking for traffic sent to the Wi-Fi network. If the array is not present but the <code>QoSmarkingKey</code> is present (even empty) no app gets whitelisted.
<code>QoSmarkingAppleVideoCalls</code>	Boolean	Optional. Specifies if audio and video traffic of built-in audio/video services such as FaceTime and Wi-Fi Calling will be whitelisted for L2 and L3 marking for traffic sent to the Wi-Fi network. Defaults to true.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

109

Key	Type	Value
<code>QoSmarkingWhitelistedAppIdentifiers</code>	Boolean	Optional. May be used to disable L2 marking and enable L3

The QoS Marking Dictionary contains these keys:

Key	Type	Value
QoS Marking Whitelist Array of App Identifiers	Strings	Optional. Array of app bundle identifiers that will be whitelisted for L2 and L3 marking for traffic sent to the Wi-Fi network. If the array is not present but the QoS Marking Key is present (even empty) no app gets whitelisted.
QoS Marking Apple Built-in Video Calls		Optional. Specifies if audio and video traffic of built-in audio/video services such as FaceTime and Wi-Fi Calling will be whitelisted for L2 and L3 marking for traffic sent to the Wi-Fi network. Defaults to true

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

109

Key	Type	Value
QoS Marking Enable Boolean		Optional. May be used to disable L3 marking and only use L2 marking for traffic sent to the Wi-Fi network. When this key is false the system behaves as if Wi-Fi was not associated with a Cisco QoS fast lane network. Defaults to true

If the Encryption field is set to WEAP or AN, the following fields may also be provided:

Key	Type	Value
Password	String	Optional.
EAP Client Configuration Dictionary		Described in EAPClientConfiguration Dictionary .
Payload Certificate String UUID		Described in Certificates .

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 Proxy field is set to Manual, the following fields must also be provided:

Key	Type	Value
Proxy Server	String	The proxy server's network address.
Proxy Server Port	Integer	The proxy server's port.
Proxy Username	String	Optional. The username used to authenticate to the proxy server.
Proxy Password	String	Optional. The password used to authenticate to the proxy server.
Proxy PAC URL	String	Optional. The URL of the PAC file that defines the proxy configuration.
Proxy PACFallback	Boolean	Optional. If false prevents the device from connecting directly to the destination if the PAC file is unreachable. Default is false

Availability: Available in iOS 7 and later.

If the Proxy field is set to Auto and no Proxy PAC URL 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 EAP Client Configuration key. If present its value is a dictionary with the following keys:

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

110

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.

Key	Type	Value
Accept EAP Types	Array of Integers	The following EAP types are accepted: 13 = TLS 17 = LEAP 18 = EAP-SIM 21 = TTLS 23 = EAP-AKA 25 = PEAP 43 = EAP-FAST For EAP-TLS authentication without a network payload, install the necessary identity certificates and have your users select EAP-TLS mode in the 802.1X credentials dialog that appears when they connect to the network. For other EAP types, a network payload is necessary and must specify the correct settings for the network.
User Password	String	Optional. User password. If not provided, the user may be prompted during login.
One Time Password	Boolean	Optional. If true the user will be prompted for a password each time they connect to the network. Defaults to false
Payload Certificate Anchor UUID	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
TLS Trusted Server 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 Payload Certificate Anchor UUID 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

TLS Trusted Server Names Optional. This is the list of server certificate common names

QoS Marking Dictionary

Optional. may be used to enable L3 marking and only use L2 marking for traffic sent to the Wi-Fi network. When this key is false the system behaves as if Wi-Fi was not associated with a Cisco QoS fast lane network. Defaults to true

If the Encryption field is set to WEAP or AN, the following fields may also be provided:

Key	Type	Value
Password	String	Optional.
EAP Client Configuration Dictionary		Described in EAPClientConfiguration Dictionary .
Payload Certificate String UUID		Described in Certificates .

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 Proxy field is set to Manual, the following fields must also be provided:

Key	Type	Value
Proxy Server	String	The proxy server's network address.
Proxy Server Port	Integer	The proxy server's port.
Proxy Username	String	Optional. The username used to authenticate to the proxy server.
Proxy Password	String	Optional. The password used to authenticate to the proxy server.
Proxy PAC URL	String	Optional. The URL of the PAC file that defines the proxy configuration.
Proxy PACFallback	Boolean	Optional. If false prevents the device from connecting directly to the destination if the PAC file is unreachable. Default is false

Availability: Available in iOS 7 and later.

If the Proxy field is set to Auto and no Proxy PAC URL 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 EAP Client Configuration key. If present its value is a dictionary with the following keys:

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

110

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.

Accept EAP Types	Array of Integers	The following EAP types are accepted: 13 = TLS 17 = LEAP 18 = EAP-SIM 21 = TTLS 23 = EAP-AKA 25 = PEAP 43 = EAP-FAST For EAP-TLS authentication without a network payload, install the necessary identity certificates and have your users select EAP-TLS mode in the 802.1X credentials dialog that appears when they connect to the network. For other EAP types, a network payload is necessary and must specify the correct settings for the network.
------------------	-------------------	--

User Password	String	Optional. User password. If not provided, the user may be prompted during login.
One Time Password	Boolean	Optional. If true the user will be prompted for a password each time they connect to the network. Defaults to false

Payload Certificate Anchor UUID	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
TLS Trusted Server 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 Payload Certificate Anchor UUID 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

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

111

OpenSSL: 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 `PayLoadCertificate` the property allows D 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`

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

111

Key	Type	Value
<code>TLSSallowTrustExcludedDomains</code>	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 <code>1</code> the authentication fails if the certificate isn't already trusted. See <code>PayLoadCertificateHash</code> and <code>TLSTrustedorNot</code> .	
<code>TLSCertificateIsRequired</code>	Optional. The default value of this property is <code>true</code> unless either <code>PayLoadCertificateHash</code> or <code>TLSTrustedorNot</code> is supplied in which case the default value is <code>false</code> .	
		Availability: Deprecated and ignored in iOS 8.0 and later.
<code>TLSCertificateIsRequired</code>	Optional. If <code>true</code> , allows for two-factor authentication for EAP-TTLS, PEAP, or EAP-FAST. If <code>1</code> allows for zero-factor authentication for EAP-TLS. The default is <code>true</code> for EAP-TLS, and <code>false</code> for other EAP types.	
		Availability: Available in iOS 7.0 and later.
<code>TLSMinimumVersionString</code>	Optional. The minimum TLS version to be used with EAP authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default minimum is 1.0.	
		Availability: Available in iOS 11.0 and macOS 10.13 and later.
<code>TLSMaximumVersionString</code>	Optional. The maximum TLS version to be used with EAP authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default maximum is 1.2.	
		Availability: Available in iOS 11.0 and macOS 10.13 and later.
<code>OuterIdentity</code>	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.
<code>TTLSInnerAuthenticationString</code>	Optional. Specifies the inner authentication used by the TTLS module. Possible values are PAP, CHAP, MSCHAP, MSCHAPv2, and EA. Defaults to MSCHAPv2.	
<code>SystemModeCredentialsStringSource</code>	Optional. On macOS, for System mode connections, this can be set to <code>ActiveDirectory</code> to use the AD machine name and password credentials.	
		Only one of the <code>SystemModeCredentialsStringSource</code> and <code>SystemModeUseOpenDirectory</code> keys should exist in the dictionary.

2010-01-22 | Copyright © 2010 Apple Inc. All Rights Reserved.

119

Key	Type	Value
<code>SystemModeUseOpenDirectoryCredentials</code>	String	<p>Optional. On macOS, in System mode, this indicates if the connection should try to use the OpenDirectory machine credentials.</p> <p>Only one of the <code>SystemModeCredentialsSource</code> and <code>SystemModeUseOpenDirectoryKeysCredentials</code> should exist in the dictionary.</p>

For information about EAR-SIM, see <https://tools-intf.nist.gov/html/refs4186>.

EAP, East Support

The EAR_EAST module uses the following properties in the EARClientConfiguration dictionary:

Key	Type	Value
E A P F A S T U s e P A C	Boolean	Optional. If true, the device will use an existing PAC if it's present. Otherwise, the server must present its identity using a certificate. Defaults to false.
E A P F A S T P r o v i s i o n P A C	Boolean	Optional. Used only if E A P F A S T U s e P A C is set to true, allows PAC provisioning. Defaults to false. This value must be set to true for EAP-FAST PAC usage to succeed, because there is no other way to provision a PAC.

Key	Type	Value
TLSSallowTrustExcludedDomains	String	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 a list the authentication fails if the certificate isn't already trusted. See Payload and Certificate each for UUID TLS Trust details.
TLSStrustedCertificates	String	The default value of this property is true unless either Payload and Certificate each for UUID TLS Trusted Server is supplied in which case the default value is false
Availability: Deprecated and ignored in iOS 8.0 and later.		
TLSCertificateIsRequired	Boolean	Optional. If true allows for two-factor authentication for EAP-TTLS, PEAP, or EAP-FAST. If it's false 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.		
TLSSMinimumVersionString	String	Optional. The minimum TLS version to be used with EAP authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default minimum is 1.0.
Availability: Available in iOS 11.0 and macOS 10.13 and later.		
TLSSMaximumVersionString	String	Optional. The maximum TLS version to be used with EAP authentication. Value may be 1.0, 1.1, or 1.2. If no value is specified, the default maximum is 1.2.
Availability: Available in iOS 11.0 and macOS 10.13 and later.		
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.
TTLSInnerAuthentication	String	Optional. Specifies the inner authentication used by the TTLS module. Possible values are PAP, CHAP, MSCHAP, MSCHAPv2, and EA. Defaults to MSCHAPv2.
SystemModeCredentialsSource	String	Optional. On macOS, for System mode connections, this can be set to ActiveDirectory to use the AD machine name and password credentials. Only one of the SystemModeCredentialsSource and SystemModeUseOpenDirectKeyCredentials should exist in the dictionary.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

11

Key	Type	Value
<code>SystemModeUseOpenStringDirectoryCredentials</code>	String	<p>Optional. On macOS, in System mode, this indicates if the connection should try to use the OpenDirectory machine credentials.</p> <p>Only one of the <code>SystemModeCredentialsSource</code>, <code>SystemModeUseOpenDirectory</code> keys/Credentials should exist in the dictionary.</p>

Note

For information about EAP-SIM, see <https://tools.ietf.org/html/rfc4186>.

EAP-Fast Support

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

Key	Type	Value
E A P F A S T U s e P A C	Boolean	Optional. If true, the device will use an existing PAC if it's present. Otherwise, the server must present its identity using a certificate. Defaults to false.
E A P F A S T P r o v i s i o n P A C	Boolean	Optional. Used only if E A P F A S T U s e P A C is set to true. Allows PAC provisioning. Default to false. This value must be set to true for EAP-FAST PAC usage to succeed, because there is no other way to provision a PAC.
E A P F A S T P r o v i s i o n P A C A n o n y m o u s l y	Boolean	Optional. If true, provisions the device anonymously. Note that there are known man-in-the-middle attacks for anonymous provisioning. Defaults to false.
E A P S I M N u m b e r O f R A N r e g e r	Integer	Optional. Number of expected RA-Ns for EAPSIM. Valid values are 2 and 3. Defaults to 3.

These keys are hierarchical in nature: if EAPFASTUsePAC is `f 1 s` the other two properties aren't consulted. Similarly, if EAPFASTProvisionPAC is `f 1 s` EAPFASTProvisionPACAnonymously isn't consulted.

If EAPFASTUsePAC is `1`, 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.

© 2019

EAPFASTProvisionPAC	Optional. If true, provisions the device anonymously. Note that there are known man-in-the-middle attacks for anonymous provisioning. Defaults to false.
EAPSIMNumberOfRands	Optional. Number of expected RANDs for EAPSIM. Valid values are 2 and 3. Defaults to 3.

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.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

113

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	UUID of the certificate payload to use for the identity credential.	

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

114

Domains Payload

This payload defines domains that are under an enterprise's management. This payload is designated by the com.apple.domains Payload Type

Unmarked Email Domains

Any email address that does not have a suffix that matches one of the unmarked email domains specified by the key Email1Dомain will be considered out-of-domain and will be highlighted as such in the Mail app.

Key	Type	Value
Email1Domains	Optional. An array of strings. An email address lacking a suffix that matches any of these strings will be considered out-of-domain.	

Managed Safari Web Domains

Opening a document originating from a managed Safari web domain causes iOS to treat the document as managed for the purpose of Managed Open In.

Key	Type	Value
WebDomains	Array	Optional. An array of URL strings. URLs matching the patterns listed here will be considered managed. Not supported in macOS

Key	Type	Value
SafariPasswordAutofillDomains	Optional. An array of URL strings. Supported in iOS 9.3 and later; not supported in macOS. Users can save passwords in Safari only from URLs matching the patterns listed here. Regardless of the iCloud account that the user is using, if the device is not supervised, there can be no whitelist. If the device is supervised, there may be a whitelist, but if there is still no whitelist, note these two cases: <ul style="list-style-type: none"> If the device is configured as Shared iPad, no password can be saved. If the device is not configured as Shared iPad, all passwords can be saved. 	

Domains Payload

This payload defines domains that are under an enterprise's management. This payload is designated by the com.apple.domains Payload Type

Unmarked Email Domains

Any email address that does not have a suffix that matches one of the unmarked email domains specified by the key Email1Dомain will be considered out-of-domain and will be highlighted as such in the Mail app.

Key	Type	Value
Email1Domains	Optional. An array of strings. An email address lacking a suffix that matches any of these strings will be considered out-of-domain.	

Managed Safari Web Domains

Opening a document originating from a managed Safari web domain causes iOS to treat the document as managed for the purpose of Managed Open In.

Key	Type	Value
WebDomains	Array	Optional. An array of URL strings. URLs matching

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

115

the patterns listed here will be considered managed. Not supported in macOS

Safari Password AutoFill Domains Optional. An array of URL strings. Supported in iOS 9.3 and later; not supported in macOS.

Users can save passwords in Safari only from URLs matching the patterns listed here.

Regardless of the iCloud account that the user is using, if the device is not supervised, there can be no whitelist. If the device is supervised, there may be a whitelist, but if there is still no whitelist, note these two cases:

- If the device is configured as Shared iPad, no password can be saved.
- If the device is not configured as Shared iPad, all passwords can be saved.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

115

The Web Domains for a Password AutoFill strings using any of the following matching patterns:

Format	Description
apple.com	Any path under apple.matches, but not site.apple.com/
foo.apple.com	Any path under foo.apple.matches but not apple.com/
bar.apple.com/	
*.apple.com	Any path under foo.apple.orbcaom.apple.matches but not apple..com
apple.com/sub.apple.com	Any path under it matches, but not apple.com/
foo.apple.com/	Any path under foo.apple.matches, but not apple..com
apple.com/	apple.com/ or sub.apple.com/
apple.com/sub	Any path under foo.apple.com/sub.apple.matches, but not apple..orbar.apple..com/
*.co	Any path under apple.orbar.apple..com/ matches, but not apple..or.uk.apple..com

A URL that begins with the prefix `www` is treated as though it did not contain that prefix during matching. For example, `http://www.apple.wdi` matches as `http://apple.com/store`

Trailing slashes will be ignored.

If a Managed Web entry contains a port number, only addresses that specify that port number will be considered managed. Otherwise, the domain will be matched without regard to the port number specified. For example, the pattern `*.apple.com:8080` will match `http://site.apple.com:8080/page.htm` and `http://site.apple.com/page.htm` will match both URLs.

Managed Safari Web Domain definitions are cumulative. Patterns defined by all Managed Web Domains payloads will be used to match a URL request.

Safari Password AutoFill Domains Optional. Patterns defined by all Safari Password AutoFill payloads will be used to determine if passwords can be stored for a given URL.

The Web Domains for a Password AutoFill strings using any of the following matching patterns:

Format	Description
apple.com	Any path under apple.matches, but not site.apple.com/
foo.apple.com	Any path under foo.apple.matches but not apple..com/
bar.apple.com/	
*.apple.com	Any path under foo.apple.orbcaom.apple.matches but not apple..com
apple.com/sub.apple.com	Any path under it matches, but not apple..com/
foo.apple.com/	Any path under foo.apple.matches, but not apple..com
apple.com/	apple.com/ or sub.apple.com/
apple.com/sub	Any path under foo.apple.com/sub.apple.matches, but not apple..orbar.apple..com/
*.co	Any path under apple.orbar.apple..com/ matches, but not apple..or.uk.apple..com

A URL that begins with the prefix `www` is treated as though it did not contain that prefix during matching. For example, `http://www.apple.wdi` matches as `http://apple.com/store`

Trailing slashes will be ignored.

If a Managed Web entry contains a port number, only addresses that specify that port number will be considered managed. Otherwise, the domain will be matched without regard to the port number specified. For example, the pattern `*.apple.com:8080` will match `http://site.apple.com:8080/page.htm` and `http://site.apple.com/page.htm` will match both URLs.

Managed Web Domain definitions are cumulative. Patterns defined by all Managed Web Domains payloads will be used to match a URL request.

Safari Password AutoFill Domains Optional. Patterns defined by all Safari Password AutoFill payloads will be used to determine if passwords can be stored for a given URL.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

116

Active Directory Payload

In macOS 10.9 and later, a configuration profile can be used to configure macOS to join an Active Directory (AD) domain. Advanced AD options available via Directory Utility or the `dsconfig` command line tool can also be set using a configuration profile, following this procedure:

1. Start with a macOS Directory payload, created in Profile Manager.

2. Save and download the profile so you can edit it manually.

The following AD configuration keys can be added to the Directory payload, of type `com.apple.DirectoryService`. Note that some settings will only be set if the associated flag key is set to `true`. For example, `ADPacketEncryptFlag` must be set to `true` to set the `ADPacketEncryptable` key.

Key	Type	Description
HostName	String	The Active Directory domain to join.
UserName	String	User name of the account used to join the domain.
Password	String	Password of the account used to join the domain.
ADOrganizationalUnit	String	The organizational unit (OU) where the joining computer object is added.
ADMountStyle	String	Network home protocol to use: "afp" or "smb".
ADCreateMobileAccountA	Boolean	Enable or disable the <code>ADCreateMobileAccountAtLogin</code> key.
ADCreateMobileAccountA	Boolean	Create mobile account at login.
ADWarnUserBeforeCreatingMobileFlag	Boolean	Enable or disable the <code>ADWarnUserBeforeCreatingMobileAccount</code> key.
ADWarnUserBeforeCreatingMobileFlag	Boolean	Warn user before creating a Mobile Account.
ADForceHomeLocalFlag	Boolean	Enable or disable the <code>ADForceHomeLocal</code> key.
ADForceHomeLocal	Boolean	Force local home directory.
ADUseWindowsUNCPathFlag	Boolean	Enable or disable the <code>ADUseWindowsUNCPath</code> key.
ADUseWindowsUNCPath	Boolean	Use UNC path from Active Directory to derive network home location.
ADAllowMultiDomainAuth	Boolean	Enable or disable the <code>ADAllowMultiDomainAuth</code> key.
ADAllowMultiDomainAuth	Boolean	Allow authentication from any domain in the forest.
ADDefaultUserShellFlag	Boolean	Enable or disable the <code>ADDefaultUserShell</code> key.
ADDefaultUserShell	String	Default user shell; e.g. <code>/bin/bash</code> .
ADMapUIDAttributeFlag	Boolean	Enable or disable the <code>ADMMapUIDAttribute</code> key.
ADMMapUIDAttribute	String	Map UID to attribute.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

116

Active Directory Payload

In macOS 10.9 and later, a configuration profile can be used to configure macOS to join an Active Directory (AD) domain. Advanced AD options available via Directory Utility or the `dsconfig` command line tool can also be set using a configuration profile, following this procedure:

1. Start with a macOS Directory payload, created in Profile Manager.

2. Save and download the profile so you can edit it manually.

The following AD configuration keys can be added to the Directory payload, of type `com.apple.DirectoryService`. Note that some settings will only be set if the associated flag key is set to `true`. For example, `ADPacketEncryptFlag` must be set to `true` to set the `ADPacketEncryptable` key.

Key	Type	Description
HostName	String	The Active Directory domain to join.
UserName	String	User name of the account used to join the domain.
Password	String	Password of the account used to join the domain.
ADOrganizationalUnit	String	The organizational unit (OU) where the joining computer object is added.

ADMountStyle	String	Network home protocol to use: "afp" or "smb".
ADCreatemobileAccountA	Boolean	Enable or disable the ADCreatemobileAccountAtLogin key.
ADCreatemobileAccountA	Boolean	Create mobile account at login.
ADWarnUserBeforeCreate	Boolean	Enable or disable the ADWarnUserBeforeCreatingMA key.
ADWarnUserBeforeCreate	Boolean	Warn user before creating a Mobile Account.
ADForceHomeLocalFlag	Boolean	Enable or disable the ADForceHomeLocal key.
ADForceHomeLocal	Boolean	Force local home directory.
ADUseWindowsUNCPathFlag	Boolean	Enable or disable the ADUseWindowsUNCPath key.
ADUseWindowsUNCPath	Boolean	Use UNC path from Active Directory to derive network home location.
ADAllowMultiDomainAuth	Boolean	Enable or disable the ADAllowMultiDomainAuth key.
ADAllowMultiDomainAuth	Boolean	Allow authentication from any domain in the forest.
ADDefaultUserShellFlag	Boolean	Enable or disable the ADDefaultUserShell key.
ADDefaultUserShell	String	Default user shell; e.g. /bin/bash.
ADMapUIDAttributeFlag	Boolean	Enable or disable the ADMapUIDAttribute key.
ADMapUIDAttribute	String	Map UID to attribute.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

117

Key	Type	Description
ADMapGIDAttributeFlag	Boolean	Enable or disable the ADMapGIDAttribute key.
ADMapGIDAttribute	String	Map user GID to attribute.
ADMapGIDAttributeFlag	Boolean	Enable or disable the ADMapGGIDAttribute key.
ADMapGGIDAttribute	String	Map group GID to attribute.
ADPreferredDCServerFlag	Boolean	Enable or disable the ADPreferredDCServer key.
ADPreferredDCServer	String	Prefers this domain server.
ADDomainAdminGroupList	Boolean	Enable or disable the ADDomainAdminGroupList key.
ADDomainAdminGroupList	Array of Strings	Allow administration by specified Active Directory groups.
ADNamespaceFlag	Boolean	Enable or disable the ADNamespace key.
ADNamespace	String	Set primary user account naming convention: "forest" or "domain"; "domain" is default.
ADPacketSignFlag	Boolean	Enable or disable the ADPacketSign key.
ADPacketSign	String	Packet signing: "allow", "disable" or "require"; "allow" is default.
ADPacketEncryptFlag	Boolean	Enable or disable the ADPacketEncrypt key.
ADPacketEncrypt	String	Packet encryption: "allow", "disable", "require" or "ssl"; "allow" is default.
ADRestrictDDNSFlag	Boolean	Enable or disable the ADRestrictDDNS key.
ADRestrictDDNS	Array of Strings	Restrict Dynamic DNS updates to the specified interfaces (e.g. en0, en1, etc.).
ADTrustChangePassInter	Boolean	Enable or disable the ADTrustChangePassInter key.
ADTrustChangePassInter	Days	How often to change the computer trust account password in days; "0" is disabled.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

118

Encrypted Profiles

A profile can be encrypted so that it can only be decrypted using a private key previously installed on a device.

To encrypt a profile do the following:

1. Remove the PayloadArray and serialize it as a proper plist. Note that the top-level object in this plist is an array, not a dictionary.
2. CMS-encrypt the serialized plist as enveloped data.
3. Serialize the encrypted data in DER format.
4. Set the serialized data as the value of as a Data plist item in the profile, using the key EncryptedPayloadContent.

Signing a Profile

To sign a profile, place the XML plist in a DER-encoded CMS Signed Data data structure.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

117

Key	Type	Description
ADMapGIDAttributeFlag	Boolean	Enable or disable the ADMapGIDAttribute key.
ADMapGIDAttribute	String	Map user GID to attribute.
ADMapGGIDAttributeFlag	Boolean	Enable or disable the ADMapGGIDAttribute key.
ADMapGGIDAttribute	String	Map group GID to attribute.
ADPreferredDCServerFlag	Boolean	Map group GID to attribute.
ADPreferredDCServer	String	Enable or disable the ADPreferredDCServer key.
ADPreferredDCServer	String	Prefers this domain server.
ADDomainAdminGroupList	Boolean	Enable or disable the ADDomainAdminGroupList key.
ADDomainAdminGroupList	Array of Strings	Allow administration by specified Active Directory groups.
ADNamespaceFlag	Boolean	Enable or disable the ADNamespace key.
ADNamespace	String	Set primary user account naming convention: "forest" or "domain"; "domain" is default.
ADPacketSignFlag	Boolean	Enable or disable the ADPacketSign key.
ADPacketSign	String	Packet signing: "allow", "disable" or "require"; "allow" is default.
ADPacketEncryptFlag	Boolean	Enable or disable the ADPacketEncrypt key.
ADPacketEncrypt	String	Packet encryption: "allow", "disable", "require" or "ssl"; "allow" is default.
ADRestrictDDNSFlag	Boolean	Enable or disable the ADRestrictDDNS key.
ADRestrictDDNS	Array of Strings	Restrict Dynamic DNS updates to the specified interfaces (e.g. en0, en1, etc.).
ADTrustChangePassInter	Boolean	Enable or disable the ADTrustChangePassInter key.
ADTrustChangePassInter	Days	How often to require change of the computer trust account password in days; "0" is disabled.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

118

Encrypted Profiles

A profile can be encrypted so that it can only be decrypted using a private key previously installed on a device.

To encrypt a profile do the following:

1. Remove the PayloadArray and serialize it as a proper plist. Note that the top-level object in this plist is an array, not a dictionary.
2. CMS-encrypt the serialized plist as enveloped data.
3. Serialize the encrypted data in DER format.
4. Set the serialized data as the value of as a Data plist item in the profile, using the key EncryptedPayloadContent.

Signing a Profile

To sign a profile, place the XML plist in a DER-encoded CMS Signed Data data structure.

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://
  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>
        </dict>
      </dict>
    </array>
  </dict>
</plist>

```

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://
  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>
        <string>0</string>
        <string>Example, Inc.</string>
      </array>
    </array>
    <array>
      <string>CN</string>
      <string>User Device Cert</string>
    </array>
  </array>
</dict>
<key>Challenge</key>
<string>...</string>
<key>Keysize</key>
<integer>1024</integer>
<key>KeyType</key>
<string>RSA</string>
<key>KeyUsage</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>
```

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

119

<array>
<array>

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

119

```
<array>
  <string>0</string>
  <string>Example, Inc.</string>
</array>
<array>
  <array>
    <string>CN</string>
    <string>User Device Cert</string>
  </array>
</array>
<key>Challenge</key>
<string>...</string>
<key>Keysize</key>
<integer>1024</integer>
<key>KeyType</key>
<string>RSA</string>
<key>KeyUsage</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>
```

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

120

```
<array>
  <string>0</string>
  <string>Example, Inc.</string>
</array>
<array>
  <array>
    <string>CN</string>
    <string>User Device Cert</string>
  </array>
</array>
<key>Challenge</key>
<string>...</string>
<key>Keysize</key>
<integer>1024</integer>
<key>KeyType</key>
<string>RSA</string>
<key>KeyUsage</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>
```

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

120

Revision History

Date	Notes
2019-01-22	Added allowPersonalHotspot and modifySoftwareUpdateDelay Restrictions Payload. Added tvOS support for forceAutomaticDateAndTime forceDelayedSoftwareUpdateDelay Restrictions Payload. Minor updates and corrections.
2018-11-07	Added the Certificate Transparency Payload.
2018-10-30	Updated the allowEnablingRestrictions restriction to refer to ScreenTime changes in iOS 12. Added allowESIMModification restriction. Added SkipTrueTypeAssistant Payload.

Revision History

Date	Notes
2019-03-25	Added macOS support for allowRemoteScreenSharing and screenShot forceClassroomAutomaticallyJoinClasses forceClassroomRequestPermissionToLeaveClasses forceClassroomUnpromptedAndDeviceLock forceClassroomUnpromptedAndDeviceLock Minor updates and corrections.
2019-01-22	Added allowPersonalHotspot and modifySoftwareUpdateDelay Restrictions Payload. Added tvOS support for forceAutomaticDateAndTime forceDelayedSoftwareUpdateDelay Restrictions Payload.

	Added a l o w C e l l u l a r P l a n to the Restrictions Payload . Added the Global Preferences Payload .
2018-09-17	Added the Mobile Accounts Payload . Minor updates and corrections. Documented the Login Items Payload . Documented macOS support for the Education Configuration Payload . Minor updates and corrections.
2018-08-27	Minor updates and corrections.
2018-08-20	Minor updates and corrections.
2018-08-06	Documented the Privacy Preferences Policy Control Pane Payload . Added a l o w M a n a g e d T o W r i t e U n m a n a g e d C o n t a c t s to the Restrictions Payload .
2018-08-06	Documented the Time Server payload. Added t o k e n R e m o v a l to the SmartCard Payload . Added A l l o w P r e R e l a s e to the Software Update Payload . Documented the ShareKit Payload deprecation; instead, use the NSExtension Management Payload .
2018-07-16	Minor updates and corrections.
2018-07-05	Added new restrictions for allow Password AutoFill , allow Password Proxim , and g a b q w e s t w o r d S h a r i n g . Updated O A u t o v a l i d i t y in the Exchange Payload . Added A l l o w A l l A p p of the SCPS Payload . Added G r o u p i n g to the Notifications Payload . Other miscellaneous updates and corrections.
2018-06-18	Converted to PDF format. Removed APN payload section. Instead, use the Cellular Payload .
2018-06-04	Updated for iOS 12, macOS 10.14, and tvOS 12.
2018-04-09	Updated for iOS 11.3, macOS 10.13.3, and tvOS 11.3.
2017-12-07	Updated for iOS 11.2, macOS 10.13.2, and tvOS 11.2 public release.
2017-09-19	Updated for iOS 11.0, macOS 10.13, and tvOS 11.0.
2017-03-27	Update for iOS 10.3.
2016-12-12	Added a link to "iOS Human Interface Guidelines" for current icon recommendations.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

121

	Restrictions Payload. Minor updates and corrections.
2018-11-07	Added the Certificate Transparency Payload .
2018-10-30	Updated the allowEnablingRestrictions restriction to refer to ScreenTime changes in iOS 12. Added allowESIMModification restriction. Added S k i p T r u e S e t t i n g to the Assistant Payload . Added a l o w C e l l u l a r P l a n to the Restrictions Payload . Added the Global Preferences Payload . Added the Mobile Accounts Payload . Minor updates and corrections.
2018-09-17	Documented the Login Items Payload . Documented macOS support for the Education Configuration Payload . Minor updates and corrections.
2018-08-27	Minor updates and corrections.
2018-08-20	Minor updates and corrections.
2018-08-06	Documented the Privacy Preferences Policy Control Pane Payload . Added a l o w M a n a g e d T o W r i t e U n m a n a g e d C o n t a c t s to the Restrictions Payload .
2018-08-06	Documented the Time Server payload. Added t o k e n R e m o v a l to the SmartCard Payload . Added A l l o w P r e R e l a s e to the Software Update Payload . Documented the ShareKit Payload deprecation; instead, use the NSExtension Management Payload .
2018-07-16	Minor updates and corrections.
2018-07-05	Added new restrictions for allow Password AutoFill , allow Password Proxim , and g a b q w e s t w o r d S h a r i n g . Updated O A u t o v a l i d i t y in the Exchange Payload . Added A l l o w A l l A p p of the SCPS Payload . Added G r o u p i n g to the Notifications Payload . Other miscellaneous updates and corrections.
2018-06-18	Converted to PDF format. Removed APN payload section. Instead, use the Cellular Payload .
2018-06-04	Updated for iOS 12, macOS 10.14, and tvOS 12.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

121

Date	Notes
2016-09-13	Made miscellaneous updates and corrections.
2016-07-01	Updated for iOS 10.0 and macOS 10.12.
2016-06-21	Added new section "Active Directory Payload"; made minor updates and corrections throughout.
2016-03-21	Updated to iOS 9.3 and made other updates and corrections.
2015-12-08	Minor updates and corrections.
2015-10-08	Minor revision.
2015-09-17	Update for iOS 9 and OS X 10.11.
2015-06-12	Made miscellaneous updates and corrections. Updated rules for removal of profiles installed through an MDM server. Added new section Network Usage Rules Payload . Added new section macOS Server Payload . Added new Email , Restrictions , SCEP , and VPN Payload keys . Clarified Web Content Filter URL matching.
2015-01-31	Added new keys to the Restrictions Payload and clarified managed domain terminology.
2014-09-17	Updated for iOS 8 and OS X v10.10.
2014-03-20	Updated for iOS 7.1.
2014-01-14	Updated for iOS 7 and OS X v10.9.
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.

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

122

Date	Notes
2018-04-09	Updated for iOS 11.3, macOS 10.13.3, and tvOS 11.3.
2017-12-07	Updated for iOS 11.2, macOS 10.13.2, and tvOS 11.2 public release.
2017-09-19	Updated for iOS 11.0, macOS 10.13, and tvOS 11.0.
2017-03-27	Update for iOS 10.3.
2016-12-12	Added a link to "iOS Human Interface Guidelines" for current icon recommendations.
2016-09-13	Made miscellaneous updates and corrections.
2016-07-01	Updated for iOS 10.0 and macOS 10.12.
2016-06-21	Added new section "Active Directory Payload"; made minor updates and corrections throughout.
2016-03-21	Updated to iOS 9.3 and made other updates and corrections.
2015-12-08	Minor updates and corrections.
2015-10-08	Minor revision.
2015-09-17	Update for iOS 9 and OS X 10.11.
2015-06-12	Made miscellaneous updates and corrections. Updated rules for removal of profiles installed through an MDM server. Added new section Network Usage Rules Payload . Added new section macOS Server Payload . Added new Email , Restrictions , SCEP , and VPN Payload keys . Clarified Web Content Filter URL matching.
2015-01-31	Added new keys to the Restrictions Payload and clarified managed domain terminology.
2014-09-17	Updated for iOS 8 and OS X v10.10.
2014-03-20	Updated for iOS 7.1.
2014-01-14	Updated for iOS 7 and OS X v10.9.
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.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

122

Copyright and Notices

Apple Inc.
Copyright © 2019 Apple Inc.
All rights reserved.

Apple Inc.
Copyright © 2019 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 or device 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-branded products.

Apple Inc.
One Apple Park Way
Cupertino, CA 95014
USA
408-996-1010

Apple is a trademark of Apple Inc., registered in the U.S. and other countries.

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, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.

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 or device 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-branded products.

Apple Inc.
One Apple Park Way
Cupertino, CA 95014
USA
408-996-1010

Apple is a trademark of Apple Inc., registered in the U.S. and other countries.

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, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.

2019-03-25 | Copyright © 2019 Apple Inc. All Rights Reserved.

123

2019-01-22 | Copyright © 2019 Apple Inc. All Rights Reserved.

123