



iOS Deployment Reference

🍏 Apple Inc.

© 2015 Apple Inc. All rights reserved.

Apple, the Apple logo, AirDrop, AirPlay, Apple TV, Bonjour, FaceTime, FileVault, iBooks, iLife, iMessage, iPad, iPad Air, iPhone, iPod, iPod touch, iTunes, iWork, Keychain, Keynote, Mac, MacBook Air, MacBook Pro, Numbers, OS X, Pages, Passbook, Safari, Siri, Spotlight, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries.

AirPrint, Apple Pay, Apple Watch, Handoff, iPad mini, iTunes U, and Touch ID are trademarks of Apple Inc.

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

iBooks Store is a service mark of Apple Inc.

The Apple logo is a trademark of Apple Inc., registered in the U.S. and other countries. Use of the “keyboard” Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Apple Inc. is under license.

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

Other company and product names mentioned herein may be trademarks of their respective companies.

Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products. All understandings, agreements, or warranties, if any, take place directly between the vendors and the prospective users.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for printing or clerical errors.

019-00124/2015-04

Contents

6	Chapter 1: iOS Deployment Reference
6	Introduction
8	Chapter 2: Deployment models
8	Overview
8	Education deployment models
8	Overview
9	Institution-owned one-to-one
11	Student-owned
13	Shared use
15	Enterprise deployment models
15	Overview
15	Personalized device (BYOD)
17	Personalized device (corporate-owned)
19	Non-personalized device (shared)
21	Chapter 3: Wi-Fi
21	Overview
21	Wi-Fi throughput
22	Join Wi-Fi
22	Roaming
23	Plan for coverage and capacity
24	Design considerations
25	Wi-Fi standards in iOS devices
27	Chapter 4: Infrastructure and integration
27	Overview
27	Microsoft Exchange
29	Bonjour
30	AirPlay
31	Standards-based services
32	Digital certificates
33	Single Sign-On (SSO)
34	Virtual private networks (VPN)
34	Overview
35	Supported protocols and authentication methods
35	SSL VPN clients
36	VPN setup guidelines
38	Per App VPN
38	VPN On Demand
38	Overview
39	Stages

39	Rules and actions
40	Backward compatibility
41	Always-on VPN
41	Overview
41	Deployment scenarios
42	Always-on VPN configuration profile
43	Always-on VPN payload
45	Chapter 5: Internet services
45	Overview
45	Apple ID
46	Find My iPhone and Activation Lock
47	Continuity
47	iCloud
48	iCloud Drive
48	iCloud Keychain
49	iMessage
49	FaceTime
49	Siri
49	Apple ID for Students
50	Apple Push Notification Service (APNs)
51	Chapter 6: Security
51	Overview
51	Device and data security
51	Overview
51	Passcode policies
52	Policy enforcement
52	Secure device configuration
52	Data protection
53	Encryption
53	Per-message S/MIME
53	External email addresses
53	Touch ID
54	Remote wipe
54	Local wipe
54	Network security
55	App security
57	Chapter 7: Configuration and management
57	Overview
57	Setup Assistant and activation
58	Configuration profiles
58	Mobile device management (MDM)
58	Overview
60	Enrollment
60	Configure
60	Accounts
61	Queries
61	Management tasks
62	Managed apps

63	Managed books
63	Managed domains
64	Profile Manager
64	Supervise devices
64	Device Enrollment Program
66	Apple Configurator
67	Chapter 8: App and book distribution
67	Overview
67	Volume Purchase Program (VPP)
67	Overview
68	Enroll in the Volume Purchase Program
68	Purchase apps and books in volume
68	Managed distribution
69	Custom B2B apps
69	In-house apps
70	In-house books
71	Deploy apps and books
71	Overview
71	Install apps and books using MDM
71	Install apps with Apple Configurator
72	Caching Server
74	Chapter 9: Planning for support
74	Overview
74	AppleCare Help Desk Support
74	AppleCare OS Support
75	AppleCare for Enterprise
75	AppleCare for iOS device users
75	iOS Direct Service Program
75	AppleCare Protection Plan for Mac or Apple Display
76	Chapter 10: Appendices
76	Restrictions
76	Overview
76	Device Enrollment Program settings
77	Device functionality
78	Supervised settings
80	Security and privacy settings
81	App usage
82	iCloud settings
82	Profile Manager user and user group restrictions
83	Install in-house apps wirelessly

iOS Deployment Reference

1

Introduction

This reference guide is for IT administrators who want to support iOS devices on their networks. It provides information about deploying and supporting iPad, iPhone, and iPod touch in a large-scale enterprise or educational organization. It explains how they provide:

- Integration with your existing infrastructure
- Comprehensive security
- Powerful tools for deployment
- Methods to distribute apps and books to your employees or students

Note: Although this reference is focused solely on the deployment of iOS devices, some sections also apply to Mac desktop and portable computers. In those instances, the term *Apple devices* will be used to represent iPhone, iPad, iPod touch, and Mac desktop and portable computers. Apple TV deployment is covered in the [AirPlay](#) section of this reference.

This guide is divided into the following sections:

Deployment models

There are several possible ways to deploy iOS devices in your organization. Regardless of the deployment model you choose, it's helpful to consider the steps you'll need to take to ensure that your deployment goes as smoothly as possible. While this guide encompasses all aspects of an iOS device deployment, businesses and educational institutions may go about their deployments differently.

Wi-Fi setup and configuration

Apple devices can securely connect to corporate or guest Wi-Fi networks right out of the box, making it quick and simple for users to join available wireless networks, whether they're on campus or on the road. This chapter discusses standard Wi-Fi protocols for data transmission and encryption.

Infrastructure and integration

iOS devices have built-in support for a wide range of network infrastructures. In this section, you'll learn about iOS-supported technologies and best practices for integrating with Microsoft Exchange, VPN, and other standard services.

Internet services

Apple has built a robust set of services to help users get the most out of their Apple devices. These services include iMessage, FaceTime, Continuity, iCloud, iCloud Keychain, and how to set up and manage Apple IDs used to access these services.

Security considerations

iOS is designed to securely access corporate services and protect important data. iOS provides strong encryption for data in transmission, proven authentication methods for access to corporate services, and hardware encryption for all data stored on iOS devices. Read this section for an overview about the security-related features of iOS.

Configuration and management

Apple devices support advanced tools and technologies to ensure that they are easily set up, configured to meet your requirements, and managed with ease in a large-scale environment. This section describes the different tools available for deployment, including an overview of mobile device management (MDM) and the Device Enrollment Program.

App and book distribution

There are a number of ways to deploy apps and content throughout your organization. Programs from Apple, including the Volume Purchase Program and the iOS Developer Enterprise Program, let your organization buy, build, and deploy apps and books for your users. Use this section to get an in-depth understanding of these programs and how to deploy apps and books purchased or built for internal use.

Planning for support

Apple provides a variety of programs and support options for users of Apple devices. Before deploying Apple devices, find out what's available for your organization and plan for any support you'll need.

The following appendices provide technical details and requirements:

MDM restrictions

Describes the restrictions you can use to configure iOS devices to meet your security, passcode, and other requirements.

Install in-house apps wirelessly

Describes how to distribute your in-house apps using your own web-based portal.

Additional resources

- www.apple.com/education/it
- www.apple.com/ipad/business/it
- www.apple.com/iphone/business/it

Note: You can find a web version of this reference at <https://help.apple.com/deployment/ios>.

Note: If the iBooks Store is available in your country or region, you can download this reference in ePub format. Simply search for *iOS Deployment Reference*.

Deployment models

2

Overview

There are several ways to distribute and set up iOS devices, from pre-configuration to employee or student self-service setup. Explore the possibilities before you get started. The tools and process you use to deploy will also be determined by your particular deployment model.

- In education, there are typically three deployment models for iOS devices: Institution-owned one-to-one, Student-owned, and Shared use. Although most institutions have a preferred model, you may encounter multiple models within your institution.
- In enterprise, there are several possible ways to deploy iOS devices in your organization. Whether you choose to deploy company-owned iOS devices, share iOS devices among employees or institute a “bring your own device” (BYOD) policy, it’s helpful to consider the steps you’ll need to take to ensure that your deployment goes as smoothly as possible.

After the deployment models are identified, your team can explore Apple’s deployment and management capabilities in detail. These tools and programs are covered extensively within this resource and should be reviewed with the key stakeholders within your organization.

Education deployment models

Overview

iPad brings an amazing set of tools to the classroom. Choosing the right strategies and tools can help transform the educational experience for teachers, students, and other users.

Whether your institution deploys iOS devices to a single classroom or across all grade levels, there are many options to easily deploy and manage iOS devices and content.

Deployment models

In educational institutions there are three common deployment models for iOS devices:

- Institution-owned one-to-one
- Student-owned
- Shared use

While most institutions have a preferred model, you may encounter multiple models within your institution.

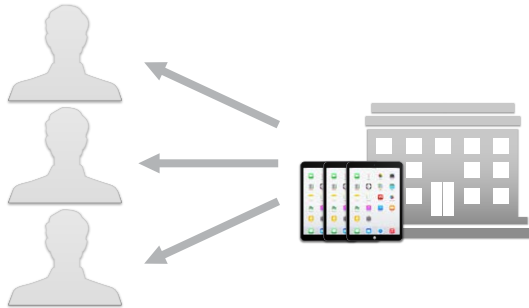
The following are a few examples of how these models would be applied in a typical education institution:

- A middle school may plan and deploy an institution-owned one-to-one model for all grade levels.
- A large district may first deploy an institution-owned one-to-one model at a single high school, and then roll out identical models for the whole district.

- A K-8 school may deploy both an institution-owned one-to-one model for fifth through eighth grades, and a shared use model for kindergarten through fourth grades.
- In higher education it is common to see the student-owned model at the campus or multi-campus levels.

Exploring these models in more detail will help you identify the best deployment model for your unique environment.

Institution-owned one-to-one



An institution-owned one-to-one deployment model provides the greatest opportunity for iOS devices to positively impact the learning process.

In a typical institution-owned one-to-one deployment, your institution purchases iOS devices for all eligible students and instructors. This could be for a particular grade level, a department, or an entire school district, college, or university.

In this model, each user is assigned an iOS device that's configured and managed by your institution. A mobile device management (MDM) solution will simplify and automate this process. If the iOS devices are purchased directly from Apple or a participating Apple Authorized Reseller or carrier, your institution can use the Device Enrollment Program (DEP) to automate enrollment in MDM, so iOS devices can be handed to users directly.

Once iOS devices are distributed, users go through a streamlined Setup Assistant, are automatically enrolled in MDM, and can further personalize their iOS device or download their own content. Users may also receive an invitation to download specific educational content, such as apps and books purchased through the Volume Purchase Program (VPP), or iTunes U courses. If students are under the age of 13, your institution can initiate the creation of an Apple ID on their behalf using the Apple ID for Students program, so apps and books can be delivered to them. Your institution can deliver or update these resources over the air anytime during the school year, and with Caching Server, most of these downloads can come from the institution's local network. If iOS devices are supervised, apps are installed automatically.

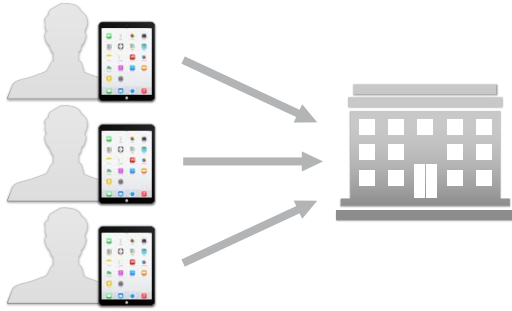
The following table illustrates the responsibilities of both the administrator and the user for an institution-owned one-to-one deployment:

Prepare	
Administrator: <ul style="list-style-type: none"> Investigate, procure, and deploy an MDM solution. Enroll in DEP, VPP, and the Apple ID for Students program. Unbox and (optionally) asset tag the iOS device. Initiate creation of Apple IDs for students under 13 (if applicable). 	Users: <ul style="list-style-type: none"> Create Apple IDs, iTunes Store, and iCloud accounts.
Set up and configure	
Administrator: <ul style="list-style-type: none"> Assign iOS devices in DEP for supervision and streamlined enrollment in MDM. Use Apple Configurator instead of DEP and MDM to configure and supervise the iOS devices. Configure and install accounts, settings, and restrictions wirelessly with MDM. 	Users: <ul style="list-style-type: none"> The user is provided a iOS device. Enter institution credentials in Setup Assistant for DEP (optional). Personalize the iOS device with Setup Assistant and enter a personal Apple ID. iOS device settings and configurations are automatically received from MDM.
Distribute devices and content	
Administrator: <ul style="list-style-type: none"> Purchase apps and books with VPP and assign them to users with MDM. Send VPP invitation to users. Install Caching Server to speed up content delivery over the local network. 	Users: <ul style="list-style-type: none"> Accept invitation to VPP. Download and install apps and books assigned by the institution. If the iOS device is supervised, apps can be pushed to the user's device silently.
Ongoing management	
Administrator: <ul style="list-style-type: none"> Revoke and reassign apps to other users as needed with MDM. With MDM, an administrator can query managed iOS devices to monitor compliance, or trigger alerts if users add unapproved apps or content. MDM can also lock iOS devices or remotely wipe any managed accounts or data, or wipe an iOS device entirely. Deploy Apple TV to support AirPlay. 	Users: <ul style="list-style-type: none"> Back up the iOS device to iTunes or iCloud, to save documents and other personal content. If the iOS device is lost or stolen, the user can locate it with Find My iPhone.

Additional Resources

- [VPP Overview](#)
- [MDM Overview](#)
- [Device Enrollment Program](#)
- [Apple ID for Students](#)
- [Apple ID](#)
- [Caching Server](#)
- [AirPlay](#)
- [Apple Configurator](#)

Student-owned



In higher education, students typically arrive on campus with their own iOS device. And while not as prevalent, in some K-12 institutions, students bring their own iOS devices to school.

In this model, iOS devices are set up and configured by the student or a parent. In order to use institutional services such as Wi-Fi, mail, and calendars, or to configure the iOS device for specific classroom requirements, student-owned iOS devices are commonly enrolled in an MDM solution provided by the institution. In education environments, technology such as MDM can play a role in managing student-owned iOS devices. Access to an institution's services acts as an incentive for users to enroll their iOS devices in the organization's MDM solution.

This ensures that all of the configuration settings, policies, restrictions, apps, books, and content are deployed automatically and unobtrusively, yet remain under the control of the institution. MDM enrollment is an "opt-in" process, so students can choose to remove management once they complete a course, graduate, or leave the institution. Removing the management also removes any content or services provided by the institution.

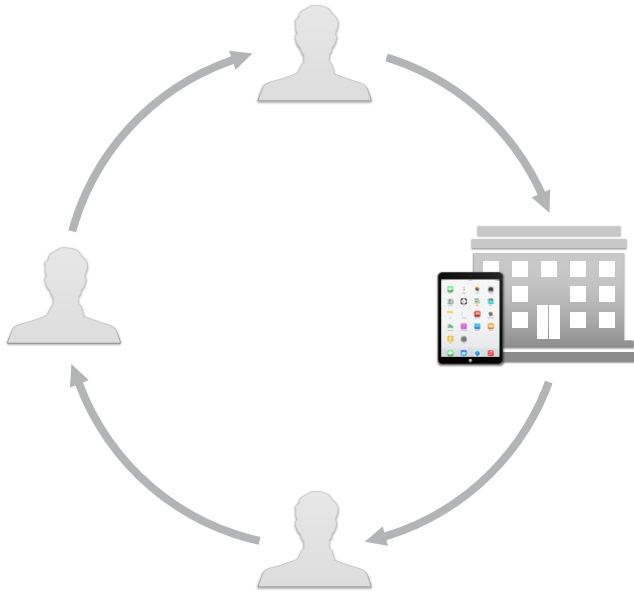
The following table illustrates the responsibilities of both the administrator and the user for a student-owned deployment:

Prepare	
Administrator: <ul style="list-style-type: none"> Investigate, procure, and deploy an MDM solution. Enroll in VPP. 	Users: <ul style="list-style-type: none"> Unbox and activate the iOS device. Create Apple ID, iTunes Store, and iCloud accounts, if applicable.
Set up and configure	
Administrator: <ul style="list-style-type: none"> No action necessary at this stage. 	Users: <ul style="list-style-type: none"> Enroll iOS devices using self service and configure accounts, settings, and restrictions wirelessly via MDM based on user/group policies defined by your institution. Personalize the iOS devices with Setup Assistant and (optionally) enter a personal Apple ID. Enroll in MDM.
Distribute apps and books	
Administrator: <ul style="list-style-type: none"> Purchase apps and books with VPP and assign them to users with MDM. Send VPP invitation to users. Install Caching Server to speed up content delivery over the local network. 	Users: <ul style="list-style-type: none"> Accept invitation to VPP. Download and install apps and books assigned by the institution. Update iOS and apps on their iOS device.
Ongoing management	
Administrator: <ul style="list-style-type: none"> Revoke and reassign apps to other users as needed with MDM. With MDM, an administrator can query managed iOS devices to monitor compliance, or trigger alerts if users add unapproved apps or content. MDM can also lock iOS devices or remotely wipe any managed accounts or data, or wipe an iOS device entirely. 	Users: <ul style="list-style-type: none"> Back up the device to iTunes or iCloud, to save documents and other personal content. If the iOS device is lost or stolen, the user can locate it with Find My iPhone. When the MDM relationship is removed, managed accounts and data are removed, but the user's personal apps, books, data, and content are kept. <p>Note: VPP books are permanently assigned. They can't be revoked.</p>

Additional Resources

- VPP [Overview](#)
- MDM [Overview](#)
- [Apple ID](#)
- [Caching Server](#)

Shared use



In a shared use model, iOS devices are purchased for use in a classroom or lab, and may be shared among students throughout the day. These devices have limited personalization, and therefore can't take full advantage of a personalized learning environment for each student. In addition to rotating devices with a shared use model, this approach could be used for a one-to-one deployment in a highly controlled context, such as a lower grade level deployment. In this case, devices have minimal personalization.

Shared use deployments are more tightly managed than personalized deployments, since the setup, configuration, and management are performed by your institution's staff. In a shared use deployment, your institution takes responsibility for installing apps, books, and other content necessary for learning.

The following table illustrates the responsibilities of both the administrator and the user for a shared use deployment:

Prepare	
Administrator: <ul style="list-style-type: none">• Investigate, procure, and deploy an MDM solution.• Enroll in VPP.• Unbox and (optionally) asset tag the iOS device.• Create institutional Apple ID(s) for each instance of Apple Configurator.	Users: <ul style="list-style-type: none">• No action necessary at this stage.
Set up and configure	
Administrator: <ul style="list-style-type: none">• Use Apple Configurator to configure and supervise devices.• Use Apple Configurator to enroll devices in MDM (optional).• Use Apple Configurator or MDM to install accounts, settings, and restrictions.	Users: <ul style="list-style-type: none">• No action necessary at this stage.
Distribute apps	
Administrator: <ul style="list-style-type: none">• Purchase apps using VPP and deploy them using redemption codes for installation and management with Apple Configurator.	Users: <ul style="list-style-type: none">• No action necessary at this stage.
Ongoing management	
Administrator: <ul style="list-style-type: none">• Update iOS on the device with Apple Configurator.• Update, configure, and install accounts, settings and restrictions wirelessly with Apple Configurator or MDM.• Periodically reset devices to standard configuration with Apple Configurator.• Install and update apps on the iOS device with Apple Configurator.• With MDM, you can query managed iOS devices to monitor compliance, or trigger alerts if users add unapproved apps or content.• MDM can also lock iOS devices or remotely wipe any managed accounts or data, or wipe an iOS device entirely.• Regular backup of the Mac running Apple Configurator is necessary, because VPP purchases are managed locally.	Users: <ul style="list-style-type: none">• No action necessary at this stage.

Additional Resources

- [VPP Overview](#)
- [MDM Overview](#)
- [Apple ID](#)
- [Apple Configurator](#)

Enterprise deployment models

Overview

iOS devices can transform your business. They can significantly boost productivity and give your employees the freedom and flexibility to work in new ways, whether in the office or on the go.

Embracing this new way of working leads to benefits across the entire organization. Users have better access to information, so they feel empowered and are able to creatively solve problems. By supporting iOS, IT departments are viewed as shaping the business strategy and solving real-world problems, rather than fixing technology and cost-cutting. Ultimately everyone benefits, with a reinvigorated workforce and new business opportunities everywhere.

Whether you're a large or small organization, there are many easy ways to deploy and manage iOS devices and content.

Start by identifying the best deployment models for your organization. Apple provides different deployment and management tools, depending on the model you choose.

Deployment models

In enterprise, there are three common deployment models for iOS devices:

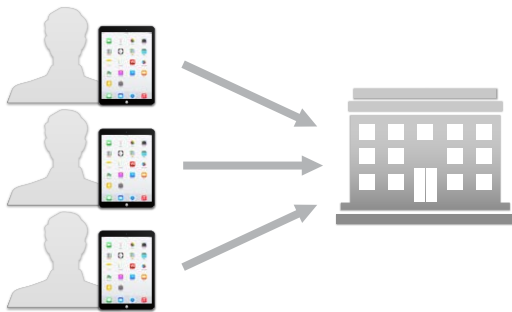
- Personalized device (BYOD)
- Personalized device (corporate-owned)
- Non-personalized (shared device)

While most organizations have a preferred model, you may encounter multiple models within your organization.

For example, a retail organization may deploy a personalized device (BYOD) strategy by allowing employees to set up their personal iPads while keeping corporate resources separate from the user's personal data and apps. However, their retail stores may also deploy a non-personalized device (shared device) strategy allowing iPod touch devices to be shared by several employees in order to process transactions for customers.

Exploring these models in more detail will help you identify the best deployment model for your unique environment.

Personalized device (BYOD)



With a bring-your-own-device deployment, users set up their personal iOS devices using their own Apple ID. In order to access corporate resources, users can configure settings manually, install a configuration profile, or more commonly, enroll the iOS device with your organization's MDM solution.

An advantage of using MDM to enroll personal iOS devices is that it keeps corporate resources separate from the user's personal data and apps. You can enforce settings, monitor corporate compliance, and remove corporate data and apps, while leaving personal data and apps on each user's iOS device.

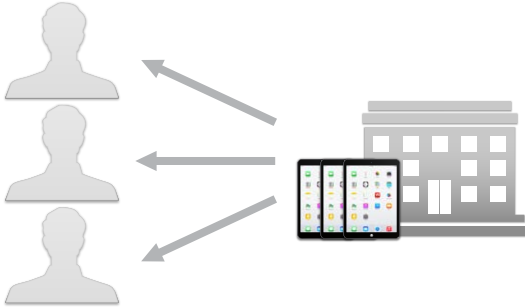
The following table illustrates the responsibilities of both the administrator and the user for a personalized device (BYOD) deployment:

Prepare	
Administrator: <ul style="list-style-type: none"> • Evaluate your existing infrastructure including Wi-Fi, VPN, and mail and calendar servers. • Investigate, procure, and deploy an MDM solution. • Enroll in VPP. 	Users: <ul style="list-style-type: none"> • Unbox and activate the iOS device. • Create Apple ID, iTunes Store, and iCloud accounts, if applicable.
Set up and configure	
Administrator: <ul style="list-style-type: none"> • Organizations can provide settings for individual accounts to users, and policies can be pushed with Exchange or installed using a configuration profile. 	Users: <ul style="list-style-type: none"> • Enroll iOS devices using self service and configure accounts, settings, and restrictions wirelessly using MDM based on user/group policies defined by your organization. • iOS device settings and configurations are automatically received from MDM. • Alternatively, users can install configuration profiles manually or configure settings as provided by you.
Distribute apps and books	
Administrator: <ul style="list-style-type: none"> • Purchase apps and books using VPP and assign them to users with MDM. • Send VPP invitation to users. • Distribute in-house apps from the iOS Developer Enterprise Program (iDEP) and in-house books by hosting them on a web server or your MDM solution. • Install Caching Server to speed up content delivery over the local network. 	Users: <ul style="list-style-type: none"> • Accept invitation to VPP. • Download and install apps and books assigned by the organization.
Ongoing management	
Administrator: <ul style="list-style-type: none"> • Revoke and reassign apps to other users as needed with MDM. • With MDM, you can query managed iOS devices to monitor compliance, or trigger alerts if users add unapproved apps or content. • MDM can also lock iOS devices or remotely wipe any managed accounts or data, or wipe an iOS device entirely. 	Users: <ul style="list-style-type: none"> • Back up the iOS device to iTunes or iCloud, to save documents and other personal content. • If the device is lost or stolen, the user can locate it with Find My iPhone. • When the MDM relationship is removed, managed accounts and data are removed, but the user's personal apps, books, data, and content are kept.

Additional Resources

- [VPP Overview](#)
- [MDM Overview](#)
- [Apple ID](#)
- [Caching Server](#)

Personalized device (corporate-owned)



You can use the personalized device model to deploy iOS devices owned by your organization. You can configure the iOS devices with basic settings before giving them to the user, or (as with BYOD) provide instructions or configuration profiles for users to apply themselves.

Alternatively, you can have users enroll their iOS devices with an MDM solution that provides settings and apps over the air. Users can then personalize the iOS devices with their own apps and data, which stay separate from your organization's managed apps and data. If the iOS devices are purchased directly from Apple or a participating Apple Authorized Reseller or carrier, your organization can use the Device Enrollment Program (DEP) to automate enrollment in MDM, so iOS devices can be handed to users directly or shipped to their homes and activated remotely.

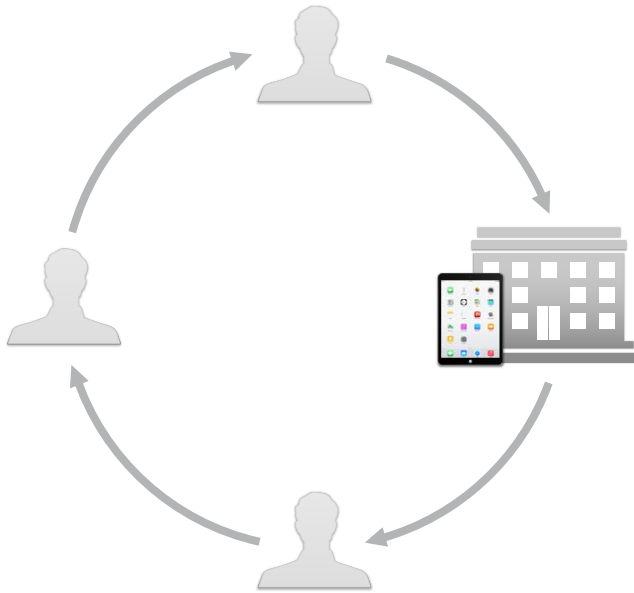
The following table illustrates the responsibilities of both the administrator and the user for a personalized device (corporate-owned) deployment:

Prepare	
Administrator: <ul style="list-style-type: none">• Evaluate your existing infrastructure including Wi-Fi, VPN, and mail and calendar servers.• Investigate, procure, and deploy an MDM solution.• Enroll in the Device Enrollment Program (DEP) and the Volume Purchase Program (VPP).	Users: <ul style="list-style-type: none">• Create Apple ID, iTunes Store, and iCloud accounts, if applicable.
Set up and configure	
Administrator: <ul style="list-style-type: none">• From the Device Enrollment Program website, link your virtual servers to your MDM solution.• Streamline enrollment through Device Enrollment Program by assigning iOS devices to your virtual MDM servers by order number or by serial number.• Assign iOS devices in DEP for supervision and streamlined enrollment in MDM.• Use Apple Configurator to configure and supervise the iOS device (alternative to the above).• Configure and install accounts, settings, and restrictions wirelessly with MDM or use USB with Apple Configurator.	Users: <ul style="list-style-type: none">• The user is provided an iOS device. If Apple Configurator was used to setup the device, then no further setup by the user is necessary.• Enter organization credentials in Setup Assistant for DEP (optional).• Personalize the iOS device with Setup Assistant and enter a personal Apple ID.• Enroll in MDM.• iOS device settings and configurations are automatically received from MDM.
Distribute apps and books	
Administrator: <ul style="list-style-type: none">• Download your token from the VPP Store and link it to your MDM solution.• Purchase apps and books using VPP and assign them to users with MDM.• Send VPP invitation to users.• Distribute in-house apps from the iOS Developer Enterprise Program (iDEP) and in-house books by hosting them on a web server or your MDM solution.• Install Caching Server to speed up content delivery over the local network.	Users: <ul style="list-style-type: none">• Accept invitation to VPP.• Download and install apps and books assigned by the organization.• If the iOS device is supervised, apps can be pushed to the user's device silently.
Ongoing management	
Administrator: <ul style="list-style-type: none">• Revoke and reassign apps to other users as needed with MDM.• With MDM, you can query managed iOS devices to monitor compliance, or trigger alerts if users add unapproved apps or content.• MDM can also lock iOS devices or remotely wipe any managed accounts or data, or wipe an iOS device entirely.	Users: <ul style="list-style-type: none">• Back up the iOS device to iTunes or iCloud, to save documents and other personal content.• If the device is lost or stolen, the user can locate it with Find My iPhone.

Additional Resources

- [VPP Overview](#)
- [MDM Overview](#)
- [Device Enrollment Program](#)
- [Apple ID](#)
- [Caching Server](#)
- [Apple Configurator](#)

Non-personalized device (shared)



If iOS devices are shared by several people or used for a single purpose (such as in a restaurant or hotel), they're typically configured and managed by you rather than by an individual user. With a non-personalized device deployment, users generally don't store personal data or have the ability to install apps.

Non-personalized devices are usually supervised with Apple Configurator and enrolled with an MDM solution. This lets the content on the device be refreshed or restored, if it's modified by a user.

The following table illustrates the responsibilities of both the administrator and the user for a non-personalized device (shared) deployment:

Prepare	
Administrator: <ul style="list-style-type: none">• Evaluate your existing infrastructure including Wi-Fi, VPN, and mail and calendar servers.• Investigate, procure, and deploy an MDM solution.• Enroll in the Volume Purchase Program (VPP).	Users: <ul style="list-style-type: none">• No action necessary at this stage.
Set up and configure	
Administrator: <ul style="list-style-type: none">• Unbox and (optionally) asset tag the iOS device.• Use Apple Configurator to configure and supervise devices.• Use Apple Configurator to enroll devices in MDM (optional).• Use Apple Configurator or MDM to install accounts, settings, and restrictions.	Users: <ul style="list-style-type: none">• No action necessary at this stage.
Distribute apps	
Administrator: <ul style="list-style-type: none">• Purchase apps using VPP and deploy them using Apple Configurator.• Distribute in-house apps from the iOS Developer Enterprise Program (iDEP) using Apple Configurator.• Distribute in-house books by hosting them on a web server or your MDM solution.	Users: <ul style="list-style-type: none">• No action necessary at this stage.
Ongoing management	
Administrator: <ul style="list-style-type: none">• Update iOS on the device with Apple Configurator.• Update, configure, and install accounts, settings and restrictions wirelessly with Apple Configurator or MDM.• Periodically reset devices to standard configuration with Apple Configurator.• Install and update apps on the device with Apple Configurator.• With MDM, you can query managed iOS devices to monitor compliance, or trigger alerts if users add unapproved apps or content.• MDM can also lock iOS devices or remotely wipe any managed accounts or data, or wipe an iOS device entirely.	Users: <ul style="list-style-type: none">• No action necessary at this stage.

Additional Resources

- [VPP Overview](#)
- [MDM Overview](#)
- [Apple ID](#)
- [Apple Configurator](#)

Overview

When preparing the Wi-Fi infrastructure for an Apple device deployment, there are several factors to consider:

- Wi-Fi throughput
- Wi-Fi trigger threshold
- Required coverage area
- Number and density of devices using the Wi-Fi network
- Types of Apple devices and their Wi-Fi capabilities
- Types and amount of data being transferred
- Security requirements for accessing the wireless network
- Encryption requirements

Although this list isn't exhaustive, it represents some of the most relevant Wi-Fi network design factors.

Note: This section focuses on Wi-Fi network design in North America. The design may differ in other countries.

Wi-Fi throughput

As you plan to deploy iOS devices within your organization, make sure your Wi-Fi network and supporting infrastructure are robust and up to date. Consistent and dependable access to a strong network is critical to setting up and configuring iOS devices. In addition, being able to support multiple iOS devices with simultaneous connections from all of your employees, students, or teachers is important to the success of your program.

Important: Your user and their iOS device must have access to your wireless network and Internet services for setup and configuration. You may need to configure your web proxy or firewall ports to allow all network traffic from Apple's devices to Apple's network (170.0.0/8), if Apple devices are unable to access Apple's activation servers, iCloud, or the iTunes Store. For a list of ports used by Apple devices, see the Apple Support article [TCP and UDP ports used by Apple software products](#).

Join Wi-Fi

Users can set Apple devices to join available Wi-Fi networks automatically. Wi-Fi networks that require login credentials or other information can be quickly accessed without opening a separate browser session, from Wi-Fi settings, or within apps such as Mail. And low-power, persistent Wi-Fi connectivity lets apps use Wi-Fi networks to deliver push notifications. You can configure settings for wireless network, security, proxy, and authentication, using configuration profiles or mobile device management (MDM).

To see how iOS decides which wireless network to auto-join, see the Apple Support article [How iOS decides which wireless network to auto-join](#).

WPA2 Enterprise

Apple devices support industry-standard wireless network protocols, including WPA2 Enterprise, ensuring corporate wireless networks can be accessed securely. WPA2 Enterprise uses 128-bit AES encryption, which assures users that their data remains protected.

With support for 802.1X, iOS devices can be integrated into a broad range of RADIUS authentication environments. 802.1X wireless authentication protocols supported by iOS include EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, IKEv2, PEAPv0, PEAPv1, and LEAP.ara.

Roaming

For roaming on large enterprise Wi-Fi networks, iOS supports 802.11k and 802.11r.

The trigger threshold is the minimum signal level a client requires to maintain the current connection.

iOS devices monitor and maintain the current BSSID's connection until the RSSI crosses the -70 dBm threshold. Once crossed, iOS initiates a scan to find roam candidate BSSIDs for the current ESSID.

This information is important to consider when designing wireless cells and their expected signal overlap. For example, if 5 GHz cells are designed with a -67 dBm overlap:

- iOS uses -70 dBm as the trigger and will therefore remain connected to the current BSSID longer than you expect.
- Review how the cell overlap was measured. The antennas on a portable computer are much larger and more powerful than a smartphone or tablet, so iOS devices see different cell boundaries than expected. It is always best to measure using the target device.

802.11k allows your iOS device to quickly identify nearby access points (AP) that are available for roaming. When the signal strength of the current AP weakens and your device needs to roam to a new AP, it will already know which AP is the best to connect with.

802.11r streamlines the authentication process using a feature called Fast Basic Service Set Transition (FT) when your iOS device roams from one AP to another on the same network. FT allows iOS devices to associate with APs more quickly. Depending on your Wi-Fi hardware vendor, FT can work with both preshared key (PSK) and 802.1X authentication methods.

Note: Not every Wi-Fi network hardware vendor supports 802.11k and 802.11r. Check with the manufacturer of your Wi-Fi hardware (controllers and APs) to find out if support is available. When you've verified support for both standards, you need to enable 802.11k and FT functionality. Setup methods vary so consult the current configuration documentation for your Wi-Fi hardware for details.

The table below shows which iOS devices can support 802.11k and 802.11r with iOS. Even if an iOS device doesn't support 802.11r, iOS 5.1 added support for "pairwise master key identifier caching" (PMKID caching), which can be used with some Cisco equipment to improve roaming between APs. Additional SSIDs might be necessary to support both FT-capable iOS devices and PMKID-caching iOS devices.

iOS device	802.11k/r support	iOS 6 and later supported methods	Pre-iOS 6 supported methods
iPad Air 2, iPad mini 3, iPhone 6, iPhone 6 Plus, iPhone 5s, iPhone 5c, iPad Air, iPad mini with Retina Display, iPad (4th generation), iPad mini, iPhone 5, iPod touch (5th generation)	Yes	FT, PMKID caching	Not applicable
iPad (3rd generation), iPhone 4s	Yes	FT, PMKID caching	PMKID caching
iPad (2nd generation) and earlier, iPhone 4 and earlier, iPod touch (4th generation) and earlier	No	PMKID caching	PMKID caching
<ul style="list-style-type: none"> • Prior to iOS 5.1, no method for optimized AP roaming existed in iOS. • "Sticky key caching" (SKC) is a form of PMKID caching. SKC is not equivalent to, nor compatible with, opportunistic key caching (OKC). 			

To view Apple's wireless roaming reference, see the Apple Support article [iOS 8: Wireless roaming reference for enterprise customers](#). For more information about roaming with 802.11k and 802.11r, see the Apple Support article [iOS: Wi-Fi network roaming with 802.11k and 802.11r](#).

Plan for coverage and capacity

Although it's important to provide Wi-Fi coverage where Apple devices are used, it's also essential to plan for the density of devices in a given area to ensure proper capacity.

Most modern, enterprise-class access points are capable of handling up to 50 Wi-Fi clients or even more, although the user experience would likely be disappointing if that many devices were actually using a single 802.11n access point. The experience for each user depends on the available wireless bandwidth on the channel the device is using, and on the number of devices sharing that bandwidth. As more devices use the same channel, the relative network speed for those devices decreases. You should consider the expected usage pattern of the Apple devices as part of your Wi-Fi network design.

Important: Avoid using hidden Service Set Identifiers (SSIDs), because Wi-Fi devices must actively seek out hidden SSIDs. This leads to delays when rejoining the SSID, potentially impacting data flow and communications. There's also no security benefit in hiding the SSID. Users tend to change location frequently along with their Apple devices, so hidden SSIDs often delay network association time and hinder roaming performance. This practice may use more power than a broadcast SSID and may affect device battery life.

2.4 GHz vs. 5 GHz

Wi-Fi networks operating at 2.4 GHz provide 11 channels in North America. However, due to channel interference considerations, only channels 1, 6, and 11 should be used in a network design.

5 GHz signals don't penetrate walls and other barriers as well as 2.4 GHz signals, which results in a smaller coverage area. Therefore, use 5 GHz networks when you design for a high density of devices in an enclosed space, such as in classrooms or large meeting rooms. The number of channels available in the 5 GHz band varies among vendors and from country to country, but at least eight channels are always available.

5 GHz channels are non-overlapping, which is a significant departure from the three non-overlapping channels available in the 2.4 GHz band. When you design a Wi-Fi network for a high density of Apple devices, the additional channels provided at 5 GHz become a strategic planning consideration.

Important: Wireless coverage should be ubiquitous throughout the workspace. If legacy devices are in use, both Wi-Fi bands—802.11b/g/n 2.4 GHz and 802.11a/n/ac 5 GHz—should be central to the design plan.

Design considerations

There are three main considerations when you design your Wi-Fi networks.

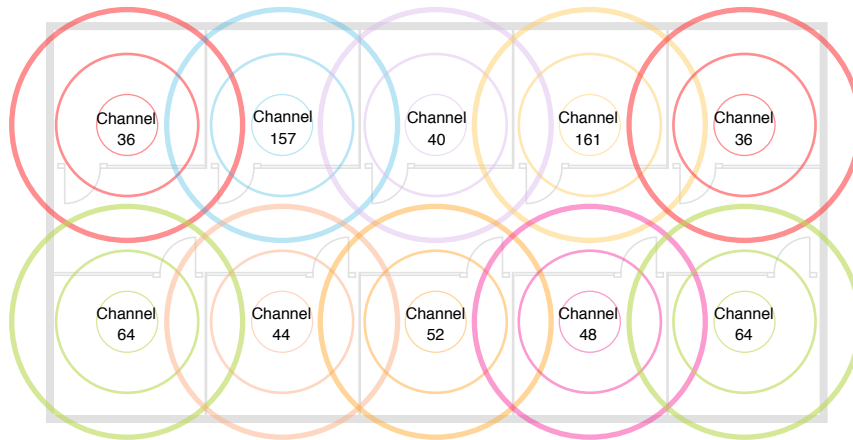
Design for coverage

The physical layout of the building may have an impact on your Wi-Fi network design. For example, in a small business environment, users may meet with other employees in conference rooms or in offices. As a result, users move around the building throughout the day. In this scenario, the majority of network access comes from low bandwidth activities such as checking mail, calendars, and Internet browsing, so Wi-Fi *coverage* is the highest priority. A Wi-Fi design could include a small number of access points on each floor to provide coverage for the offices. Additional access points might be considered for areas where large numbers of employees gather, such as a large conference room.

Design for capacity

Contrast the scenario above with a school that has 1000 students and 30 teachers in a two-story building. Every student is issued an iPad, and every teacher is issued both a MacBook Air and an iPad. Each classroom holds approximately 35 students, and classrooms are next to each other. Throughout the day, students conduct research on the Internet, watch curriculum videos, and copy files to and from a file server on the Local Area Network (LAN).

The Wi-Fi network design for high density is more complex, due to the higher density of mobile devices. Because of the large number of devices in each classroom, one access point per classroom might be required. Multiple access points should be considered for the common areas, to provide adequate *coverage* and *capacity*. The number of access points for the common areas may vary, depending on the density of Wi-Fi devices in those spaces.



Important: A pre-install site survey should always be performed to determine the exact number of access points needed and where those access points should be mounted. A site survey also determines the proper power settings for each radio. Once the installation of the Wi-Fi network is complete, a post-install site survey should be performed to confirm the Wi-Fi environment. For example, if designing a network to support a large number of people in a building it is best to validate the design with people in the building. (Or, if classroom doors will be closed when the network is in use, the door should be closed when validating the design as well.)

Sometimes it is desirable to create multiple SSIDs for different purposes. For example, a guest network may be required. Care should be taken to avoid creating too many SSIDs, as the additional SSIDs cause additional bandwidth usage.

Design for Applications

Apple products use multicast networking for services like AirPlay and AirPrint. Therefore, multicast support should be part of the design plan. For information about how to prepare your network for Bonjour, see [Bonjour](#).

Wi-Fi standards in iOS devices

Wi-Fi specifications for Apple iOS devices are detailed in the list that follows, which includes this information:

- **802.11 compatibility:** 802.11ac, 802.11n, 802.11a, 802.11b/g
- **Frequency band:** 2.4 GHz or 5 GHz
- **Maximum Transmit Rate:** This is the highest rate at which a client can transmit data over Wi-Fi.
- **Spatial streams:** Each radio can send independent data streams at the same time, each containing different data, which can increase overall throughput. The number of these independent data streams is defined as the number of spatial streams.
- **MCS index:** The Modulation and Coding Scheme (MCS) index defines the maximum transmission rate at which 802.11ac/n devices can communicate. 802.11ac uses Very High Throughput (VHT) and 802.11n uses High Throughput (HT).

- *Channel width*: The maximum channel width. Beginning with 802.11n, channels can be combined to create a wider channel that allows for more data to be transmitted during a single transmission. With 802.11n, two 20 MHz channels can be combined to create a 40 MHz channel. With 802.11ac, four 20 MHz channels can be combined to create an 80 MHz channel.
- *Guard interval (GI)*: The guard interval is the space (time) between symbols transmitted from one device to another. The 802.11n standard defines the option of a short guard interval of 400ns that provides faster overall throughput, but devices may use a long guard interval of 800ns.

Model	802.11 compatibility and frequency band	Maximum Transmit Rate	Spatial streams	MCS index	Channel width	Guard interval
iPad Air 2	802.11ac/n/a @ 5 GHz 802.11 n/g/b @ 2.4 GHz	866 Mbps	2	9 (VHT) 15 (HT)	80 MHz	400 ns
iPad mini 3	802.11n @ 2.4 GHz and 5GHz 802.11a/b/g	300 Mbps	2	15 (HT)	40 MHz	400 ns
iPhone 6 Plus iPhone 6	802.11ac/n/a @ 5 GHz 802.11 n/g/b @ 2.4 GHz	433 Mbps	1	9 (VHT) 7 (HT)	80 MHz	400 ns
iPhone 5s iPhone 5c iPhone 5	802.11n @ 2.4 GHz and 5GHz 802.11a/b/g	150 Mbps	1	7 (HT)	40 MHz	400 ns
iPhone 4s iPhone 4	802.11n @ 2.4GHz 802.11b/g	65 Mbps	1	7 (HT)	20 MHz	800 ns
iPad Air iPad mini with Retina display	802.11n @ 2.4GHz and 5GHz 802.11a/b/g	300 Mbps	2	15 (HT)	40 MHz	400 ns
iPad (4th generation) iPad mini	802.11n @ 2.4GHz and 5GHz 802.11a/b/g	150 Mbps	1	7 (HT)	40 MHz	400 ns
iPad (1st, 2nd, and 3rd generation)	802.11n @ 2.4GHz and 5GHz 802.11a/b/g	65 Mbps	1	7 (HT)	20 MHz	800 ns
iPod touch (5th generation)	802.11n @ 2.4GHz and 5GHz 802.11a/b/g	150 Mbps	1	7 (HT)	40 MHz	400 ns
iPod touch (4th generation)	802.11n @ 2.4GHz 802.11b/g	65 Mbps	1	7 (HT)	20 MHz	800 ns

Infrastructure and integration

4

Overview

iOS supports a wide range of network infrastructures, including the following:

- Local networking using Bonjour
- Cable-free connections to Apple TV using AirPlay
- Digital certificates to authenticate users and secure communications
- Single Sign-On to streamline authentication to networked apps and services
- Standards-based mail, directory, calendar, and other systems
- Popular third-party systems like Microsoft Exchange
- Virtual private networks (VPN), including Per App VPN and Always-on VPN

This support is built into iOS, so your IT department needs to configure only a few settings to integrate iOS devices into your existing infrastructure. Read on to learn more about iOS-supported technologies and guidelines for business and education.

Microsoft Exchange

iOS can communicate directly with your Microsoft Exchange Server using Microsoft Exchange ActiveSync (EAS), enabling push email, out-of-office replies, calendar, contacts, notes, and tasks. Exchange ActiveSync also provides users with access to the Global Address List (GAL), and provides administrators with passcode policy enforcement and remote wipe capabilities. iOS supports both basic and certificate-based authentication for Exchange ActiveSync.

If your organization currently uses Exchange ActiveSync, you have the necessary services in place to support iOS—no additional configuration is necessary.

Requirements

iOS 8 or later supports the following versions of Microsoft Exchange:

- Office 365 (using EAS 14.1)
- Exchange Server 2013 (using EAS 14.1)
- Exchange Server 2010 SP 2 (using EAS 14.1)
- Exchange Server 2010 SP 1 (EAS 14.1)
- Exchange Server 2010 (EAS 14.0)
- Exchange Server 2007 SP 3 (EAS 12.1)
- Exchange Server 2007 SP 2 (EAS 12.1)
- Exchange Server 2007 SP 1 (EAS 12.1)
- Exchange Server 2007 (using EAS 2.5)

Microsoft Exchange Autodiscovery

iOS and OS X support the Autodiscover service of Microsoft Exchange Server 2007 or later. When you manually configure an Apple device, Autodiscover uses your email address and password to determine the correct Exchange Server information.

For more information, see [Autodiscover Service](#) at the Microsoft website.

Microsoft Direct Push

Exchange Server automatically delivers email, tasks, contacts, and calendar events to iOS devices, provided a cellular or Wi-Fi data connection is available.

Microsoft Exchange Global Address List (GAL)

Apple devices retrieve contact information from your organization's Exchange Server corporate directory. You can access the directory while searching in Contacts, and it's automatically accessed for completing email addresses as you enter them.

Note: iOS 6 or later supports GAL photos (requires Exchange Server 2010 SP 1 or later).

Set out-of-office reply message

iOS 8 supports the use of automatic message replies when the user is unavailable. The user can also select an end date for the replies.

Calendar

iOS 8 or later and OS X Mavericks or later support the following features of Microsoft Exchange:

- Wirelessly create and accept calendar invitations
- View an invitee's calendar free/busy information
- Create private calendar events
- Configure custom repeating events
- View the week numbers in Calendar
- Receive calendar updates
- Sync tasks with the Reminders app

View the Exchange identifier

iOS 8 lets the user see the unique device identifier that's seen by the Exchange Server, called the Exchange Device ID. This is useful when the Exchange Server the user connects to requires devices to be whitelisted before access is allowed. They can supply this identifier to you in advance. The Exchange Device ID changes only if the device is restored back to factory settings. It won't change when upgrading from iOS 7 to iOS 8. To view the Exchange Device ID on an iOS device, tap Settings > Mail, Contacts, Calendars > Add Account, then tap Exchange.

Identify iOS versions with Exchange

When an iOS device connects to an Exchange Server, the device reports its iOS version. The version number is sent in the User Agent field of the request header, and looks like Apple-iPhone2C1/705.018. The number after the delimiter (/) is the iOS build number, which is unique to each iOS release.

To view the build number on a device, go to Settings > General > About. You'll see the version number and build number, such as 4.1 (8B117A). The number in parentheses is the build number, which identifies the release the device is running.

When the build number is sent to the Exchange Server, it's converted from the format *NANNNNA* (where *N* is numeric and *A* is an alphabetic character) to the Exchange format *NNN.NNN*. Numeric values are kept, but letters are converted to their position value in the alphabet. For example, "F" is converted to "06" because it's the sixth letter in the alphabet. Numbers are padded with zeros if necessary, to fit the Exchange format. In this example, the build number 7E18 is converted to "705.018."

The first number, 7, remains as "7." The character E is the fifth letter in the alphabet so it's converted to "05." A period (.) is inserted in the converted version, as required by the format. The next number, 18, is padded with zero and converted to "018."

If the build number ends with a letter, such as 5H11A, the number is converted as described above, and the numeric value of the final character is appended to the string, separated by 3 zeroes. So 5H11A becomes "508.01100001."

Remote Wipe

You can remotely wipe the contents of an iOS device using features provided by Exchange. Wiping removes all data and configuration information from the device, and the device is securely erased and restored to its original factory settings. Wiping removes the encryption key to the data (encrypted using 256-bit AES encryption), which immediately makes all of the data unrecoverable.

With Microsoft Exchange Server 2007 or later, you can perform a remote wipe with the Exchange Management Console, Outlook Web Access, or the Exchange ActiveSync Mobile Administration Web Tool. With Microsoft Exchange Server 2003, you can initiate a remote wipe using the Exchange ActiveSync Mobile Administration Web Tool.

Alternatively, users can wipe their own device by going to Settings > General > Reset and choosing "Erase All Content and Settings." Devices can also be configured to be automatically wiped after a specified number of failed passcode attempts.

Bonjour

Bonjour is Apple's standards-based, zero configuration network protocol that lets devices find services on a network. iOS devices use Bonjour to discover AirPrint-compatible printers, and both iOS devices and Mac computers use Bonjour to discover AirPlay-compatible devices such as Apple TV. Some apps also use Bonjour for peer-to-peer collaboration and sharing.

Bonjour works by using multicast traffic to advertise the availability of services. Multicast traffic is usually not routed, so make sure Apple TV devices or AirPrint printers are on the same IP subnet as the iOS devices that would use them. If your network is larger and utilizes many IP subnets, you may want to consider using a Bonjour gateway such as those offered by various Wi-Fi infrastructure manufacturers.

For more information about Bonjour, see Apple's [Bonjour](#) webpage and Apple's Developer documentation on [Bonjour](#).

AirPlay

iOS 8 and OS X Yosemite support the ability to stream content from an Apple device to Apple TV even if the devices are on different networks or there's no network available. The Apple device uses Bluetooth® Low Energy (BTLE) to begin the discovery process of available Apple TV devices and then establishes a connection directly to Apple TV using Wi-Fi. Bluetooth Low Energy discovery is a distinct subset of peer-to-peer AirPlay.

In iOS 8 and OS X Yosemite, peer-to-peer AirPlay lets a user use AirPlay directly from a supported iOS device or Mac to an Apple TV without first connecting to an organization's network. Peer-to-peer AirPlay eliminates the need to join the right network or disclose Wi-Fi passwords, avoids reachability issues in complex network environments, and provides a direct path from the AirPlay sender to AirPlay receiver to optimize performance. Peer-to-peer AirPlay is enabled by default in iOS 8 and OS X Yosemite, and doesn't require any user configuration.

Peer-to-peer AirPlay requires:

- Apple TV (3rd generation rev A Model A1469 or later) with Apple TV software 7.0 or later
- iOS devices (late 2012 or later) with iOS 8 or later
- Mac computers (2012 or later) with OS X Yosemite or later

To find the model number of an Apple TV, see the Apple Support article [Identifying Apple TV models](#).

Peer-to-Peer discovery is initiated using Bluetooth Low Energy (BTLE) when a user selects AirPlay on an iOS device running iOS 8 or Mac running OS X Yosemite. This causes the device and the Apple TV to visit Wi-Fi channel 149,1 in the 5 GHz band and Wi-Fi channel 6 in the 2.4 GHz band, where the discovery process continues. Once the user selects an Apple TV and AirPlay starts, the Wi-Fi radios timeshare between channel 149,1 and whichever infrastructure channel each device is currently using. If possible, the AirPlay sender roams to the same infrastructure channel the Apple TV is using. If neither device is currently using an infrastructure network, the devices will utilize Wi-Fi channel 149 *only* for AirPlay. Peer-to-peer AirPlay adheres to 802.11 standards, sharing Wi-Fi bandwidth with other Wi-Fi devices.

When you deploy Apple TVs on a large enterprise Wi-Fi network, consider the following guidelines:

- Connect Apple TV to Ethernet whenever possible.
- If possible, avoid using Wi-Fi Channels 149 and 153 for your infrastructure network.
- Don't place or mount the Apple TV behind objects that could disrupt the Bluetooth Low Energy and Wi-Fi signals.
- When mounting an Apple TV to a wall or other surface, always mount it with the foot side to the surface.
- If peer-to-peer AirPlay isn't supported on either the AirPlay sender or receiver, then the infrastructure connection is automatically used.

AirPlay discovery

iOS devices will continue to use the same discovery methods available today to find AirPlay receivers. AirPlay receivers can advertise themselves using Bonjour or Bluetooth. Discovery over Bluetooth requires iOS 7.1 or later on the following:

- iPad Air
- Apple TV (3rd generation or later) running software 6.1 or later
- iPhone 4s or later

- iPad 3rd generation or later
- iPad mini 1st generation or later
- iPod touch 5th generation or later

Discovered AirPlay receivers appear in the AirPlay menu.

Bonjour services `_airplay._tcp` and `_raop._tcp` need to be advertised on Bonjour gateway products. Contact your gateway vendor to make sure these services are advertised.

Connectivity

Infrastructure and peer-to-peer are the two supported modes of AirPlay connectivity. If both the AirPlay sender and receiver support peer-to-peer AirPlay, that's the preferred data path regardless of infrastructure availability. Peer-to-peer AirPlay coexists with infrastructure connections, so the AirPlay client or AirPlay sender can maintain Internet connectivity simultaneously with the peer-to-peer connection. The 5 GHz band is better for connecting over peer-to-peer AirPlay, because it provides a fast, direct connection between the AirPlay sender and AirPlay receiver.

Security

AirPlay uses AES encryption to ensure that content remains protected when mirroring or streaming from an iOS device or Mac to an Apple TV.

AirPlay access to an Apple TV can be restricted by setting an Onscreen Code or Password. Only users who enter the Onscreen Code (per AirPlay attempt) or Password on their iOS device or Mac can send AirPlay content to an Apple TV.

Enabling Require Device Verification (Requires an iOS device with iOS 7.1 or later or a Mac with OS X Mavericks v/10.9.2 or later.) requires the iOS device or Mac to authenticate on the initial AirPlay connection. Require Device Verification is useful when Apple TV is deployed on an open Wi-Fi network. To ensure iOS devices and Mac computers are securely paired, the user is prompted to enter in a one-time Onscreen code. Subsequent connections don't require a code, unless Onscreen Code settings are enabled. Restoring an Apple TV or a previously-paired client to factory settings resets the initial connection condition.

Peer-to-peer AirPlay is always secured with Require Device Authentication. This setting isn't configurable by the user, and it prevents any nearby unauthorized users from accessing an Apple TV.

Note: For devices not on an infrastructure network, Bonjour advertisement of supported AppleTV devices (A1469 or later) is triggered by Bluetooth.

Standards-based services

With support for the IMAP mail protocol, LDAP directory services, CalDAV calendaring, and CardDAV contacts protocols, iOS and OS X can integrate with just about any standards-based environment. And if your network environment is configured to require user authentication and SSL, iOS and OS X provide a secure approach to accessing standards-based corporate email, calendar, tasks, and contacts. With SSL, iOS and OS X support 128-bit encryption and X.509 root certificates issued by the major certificate authorities.

In a typical deployment, Apple devices establish direct access to IMAP and SMTP mail servers to send and receive mail over the air (or in the case of a Mac, over the air or Ethernet), set VIP status in their message threads, and can also wirelessly sync notes with IMAP-based servers. Apple devices can connect to your organization's LDAPv3 corporate directories, giving users access to corporate contacts in the Mail, Contacts, and Messages apps. CardDAV support lets your users maintain a set of contacts synced with your CardDAV server using the vCard format. Synchronization with your CalDAV server lets users do the following:

- Create and accept calendar invitations
- View an invitee's calendar free/busy information
- Create private calendar events
- Configure custom repeating events
- View the week numbers in Calendar
- Receive calendar updates
- Sync tasks with the Reminders app

All network services and servers can be within a DMZ subnetwork, behind a corporate firewall, or both.

Digital certificates

Apple devices support digital certificates and identities, giving your organization streamlined access to corporate services. These certificates can be used in a variety of ways. For example, the Safari browser can check the validity of an X.509 digital certificate and set up a secure session with up to 256-bit AES encryption. This involves verifying that the site's identity is legitimate and that communication with the website is protected to help prevent interception of personal or confidential data. Certificates can also be used to guarantee the identity of the author or "signer" and can be used to encrypt mail, configuration profiles, and network communications to further protect confidential or private information.

Use certificates with Apple devices

Out of the box, Apple devices include a number of preinstalled root certificates from various Certification Authorities (CA) and iOS validates the trust for these root certificates. If iOS can't validate the trust chain of the signing CA, the service will encounter an error. For example, a self-signed certificate can't be verified by default in iOS. To view the current list of trusted root certificates in iOS, see the Apple Support article [iOS 8: List of available trusted root certificates](#).

iOS devices can update certificates wirelessly, if any of the preinstalled root certificates become compromised. To disable this, there's an MDM restriction that prevents over-the-air certificate updates.

These digital certificates can be used to securely identify a client or server, and encrypt the communication between them utilizing the public and private key pair. A certificate contains a public key, information about the client (or server), and is signed (verified) by a CA.

A certificate and its associated private key are known as an *identity*. Certificates can be freely distributed, but identities must be kept secured. The freely distributed certificate, and especially its public key part, are used for encryption that can be decrypted only by the matching private key. To secure the private key of an identity, it is stored in a PKCS12 file, encrypted with another key that is protected by a passphrase. An identity can be used for authentication (such as 802.1x EAP-TLS), signing, or encryption (such as S/MIME).

The list of supported certificate and identity formats on Apple devices are:

- X.509 certificates with RSA keys
- *Certificate*: .cer, .crt, .der
- *Identity*: .pfx, .p12

Deploy certificates to establish trust with Certification Authorities (CA) that are not trusted by default (such as an organizational-issuing certification authority).

Distribute and install certificates

Manually distributing certificates to iOS devices is simple. When a certificate is received, users simply tap to review the contents, then tap to add the certificate to their device. When an identity certificate is installed, users are prompted for the password that protects it. If a certificate's authenticity can't be verified, it's shown as untrusted and the user can decide whether to add it to their device.

Install certificates using configuration profiles

If configuration profiles are being used to distribute settings for corporate services such as S/MIME mail, VPN, or Wi-Fi, certificates can be added to the profile to streamline deployment. This includes the ability to distribute certificates with MDM.

Install certificates via Mail or Safari

If a certificate is sent in a mail message, it appears as an attachment. Safari can also be used to download certificates from a webpage. You can host a certificate on a secured website and provide users with the URL where they can download the certificate onto their Apple device.

Certificate removal and revocation

To manually remove a certificate that's been installed, choose Settings > General > Device Management, select a profile, choose More Details, and choose the appropriate certificate to remove. If a user removes a certificate that's required for accessing an account or network, the iOS device is no longer able to connect to those services.

An MDM server can view all certificates on a device and remove any certificates it has installed.

Additionally, the Online Certificate Status Protocol (OCSP) and CRL (Certificate Revocation List) protocol are supported to check the status of certificates. When an OCSP- or CRL-enabled certificate is used, both iOS and OS X periodically validate it to make sure that it hasn't been revoked.

Single Sign-On (SSO)

Single Sign-On (SSO) is a process in which a user can provide authentication information once, receive a ticket, and use it to access resources for as long as the ticket is valid. This strategy makes it possible to maintain secure access to resources without the system prompting the user for credentials every time access is requested. It also increases the security of daily app use, by ensuring that passwords are never transmitted over the network.

With iOS 7 or later, apps can take advantage of your existing in-house Single Sign-On infrastructure via Kerberos. The Kerberos authentication system used by iOS 7 or later is the most commonly deployed Single Sign-On technology in the world. If you have Active Directory, eDirectory, or Open Directory, it's likely to already have a Kerberos system in place that iOS 7 or later can use. iOS devices need to be able to contact the Kerberos service over a network connection to authenticate users. In iOS 8, certificates can be used to silently renew a Kerberos ticket, letting users maintain connections to certain services that leverage Kerberos for authentication.

Supported apps

iOS provides flexible support for Kerberos Single Sign-On to any app that uses the `NSURLConnection` or `NSURLSession` class to manage network connections and authentication. Apple provides all developers with these high-level frameworks to make network connections seamlessly integrated within their apps. Apple also provides Safari, as an example to help you get started by using SSO-enabled websites natively.

Configure Single Sign-On

You configure Single Sign-On using configuration profiles, which may be either manually installed or managed with MDM. The Single Sign-On payload allows flexible configuration. Single Sign-On can be open to all apps, or restricted by app identifier, service URL, or both.

Simple pattern matching is used for URLs which must begin with either `http://` or `https://`. The matching is on the entire URL, so be sure that they're exactly the same. For example, a `URLPrefixMatches` value of `https://www.example.com/` won't match `https://www.example.com:443/`. You may specify `http://` or `https://` to restrict the use of SSO to either secure or regular HTTP services. For example, using a `URLPrefixMatches` value of `https://` allows the SSO account to be used only with secure HTTPS services. If a URL matching pattern doesn't end with a slash (/), a slash is appended.

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

Virtual private networks (VPN)

Overview

Secure access to private corporate networks is available in iOS and OS X using established industry-standard virtual private network (VPN) protocols. Out of the box, iOS and OS X support Cisco IPsec, L2TP over IPsec, and PPTP. iOS also supports IKEv2. If your organization supports one of these protocols, no additional network configuration or third-party apps are required in order to connect Apple devices to your VPN.

iOS and OS X support SSL VPN from popular VPN providers. Like other VPN protocols supported in iOS and OS X, SSL VPN can be configured manually on the Apple device, or by configuration profiles or mobile device management.

iOS and OS X also support industry-standard technologies such as IPv6, proxy servers, and split-tunneling, providing a rich VPN experience when connecting to corporate networks. And iOS and OS X work with a variety of authentication methods including password, two-factor token, digital certificates, and for OS X, Kerberos. To streamline the connection in environments where certificate-based authentication is used, iOS and OS X feature VPN On Demand, which initiates a VPN session when it's needed in order to connect to specified domains.

With iOS 7 or later and OS X Yosemite or later, individual apps can be configured to use a VPN connection independent from other apps. This ensures that corporate data always flows over a VPN connection, and other data, such as an employee's personal apps from the App Store, does not. For details, see [Per App VPN](#).

iOS also features Always-on VPN, when an iOS device must connect to a known, approved VPN before connecting to any other network services. You can configure Always-on VPN for both cellular and Wi-Fi configurations. For example, using Always-on VPN, an iOS device must connect to a known and approved VPN before connecting to any other network services such as mail, web, or messages. This feature depends on your VPN provider supporting this configuration, and is available only for supervised devices. For information, see the Always-on VPN [Overview](#).

Supported protocols and authentication methods

iOS and OS X support the following protocols and authentication methods:

- *L2TP over IPSec*: User authentication by MS-CHAP v2 password, two-factor token, certificate, machine authentication by shared secret or certificate.
- *SSL VPN*: User authentication by password, two-factor token, certificates using a third-party VPN client.
- *Cisco IPSec*: User authentication by password, two-factor token, machine authentication by shared secret and certificates.
- *IKEv2*: Certificates (RSA-only), EAP-TLS, EAP-MSCHAPv2. (iOS-only)
- *PPTP*: User authentication by MS-CHAP v2 password, certificate, and two-factor token.

OS X can also use Kerberos machine authentication by shared secret or certificate with L2TP over IPSec and with PPTP.

SSL VPN clients

Several SSL VPN providers have created apps to help configure iOS devices for use with their solutions. To configure a device for a specific solution, install the companion app from the App Store and, optionally, provide a configuration profile with the necessary settings.

SSL VPN solutions include:

- *AirWatch SSL VPN*: For information, see the [AirWatch](#) website.
- *Aruba Networks SSL VPN*: iOS supports Aruba Networks Mobility Controller. For configuration, install the Aruba Networks VIA app, available on the App Store.
For contact information, see the [Aruba Networks](#) website.
- *Check Point Mobile SSL VPN*: iOS supports the Check Point Security Gateway with a full Layer-3 VPN tunnel. Install the Check Point Mobile app, available on the App Store.
- *Cisco AnyConnect SSL VPN*: iOS supports Cisco Adaptive Security Appliance (ASA) running suggested software release 8.2.5 or later. Install the Cisco AnyConnect app, available on the App Store.
- *F5 SSL VPN*: iOS supports F5 BIG-IP Edge Gateway, Access Policy Manager, and FirePass SSL VPN solutions. Install the F5 BIG-IP Edge Client app, available on the App Store.

For more information, see the F5 technical brief [Secure iPhone Access to Corporate Web Applications](#).

- *Juniper Junos Pulse SSL VPN*: iOS supports Juniper Networks SA Series SSL VPN Gateway running version 6.4 or later with Juniper Networks IVE package 7.0 or later. Install the Junos Pulse app, available on the App Store.

For more information, see [Junos Pulse](#) on the Juniper Networks website.

- *Mobile Iron SSL VPN*: For information, see the [Mobile Iron](#) website.
- *NetMotion SSL VPN*: For information, see the [NetMotion](#) website.
- *OpenVPN SSL VPN*: iOS supports OpenVPN Access Server, Private Tunnel, and OpenVPN Community. For configuration, install the OpenVPN Connect app, available on the App Store.
- *Palo Alto Networks GlobalProtect SSL VPN*: iOS supports the GlobalProtect gateway from Palo Alto Networks. Install the GlobalProtect for iOS app, available on the App Store.
- *SonicWALL SSL VPN*: iOS supports SonicWALL Aventail E-Class Secure Remote Access appliances running 10.5.4 or later, SonicWALL SRA appliances running 5.5 or later, and SonicWALL Next-Generation Firewall appliances including the TZ, NSA, E-Class NSA running SonicOS 5.8.1.0 or later. Install the SonicWALL Mobile Connect app, available on the App Store.

For more information, see the [SonicWALL](#) website.

VPN setup guidelines

Cisco IPSec setup guidelines

Use these guidelines to configure your Cisco VPN server for use with iOS devices. iOS supports Cisco ASA 5500 Security Appliances and PIX Firewalls configured with 7.2.x software or later. The latest software release (8.0.x or later) is recommended. iOS also supports Cisco IOS VPN routers with IOS version 12.4(15)T or later. VPN 3000 Series Concentrators don't support iOS VPN capabilities.

Proxy setup

For all configurations, you can specify a VPN proxy:

- To configure a single proxy for all connections, use the Manual setting and provide the address, port, and authentication if necessary.
- To provide the device with an auto-proxy configuration file using PAC or WPAD, use the Auto setting. For PACS, specify the URL of the PACS or JavaScript file. For WPAD, iOS asks DHCP and DNS for the appropriate settings.

The VPN proxy configuration gets used when the VPN is providing the following:

- *The default resolver and the default route*: The VPN proxy is used for all web requests on the system.
- *A split tunnel*: Only connections to hosts that match the VPN's DNS search domains will use the VPN proxy.

Authentication methods

iOS supports the following authentication methods:

- Pre-shared key IPSec authentication with user authentication via xauth.
- Client and server certificates for IPSec authentication, with optional user authentication via xauth.
- Hybrid authentication, where the server provides a certificate and the client provides a pre-shared key for IPSec authentication. User authentication is required via xauth.
- User authentication is provided via xauth and includes the following authentication methods:

- Username with password
- RSA SecurID
- CRYPTOCARD

Authentication groups

The Cisco Unity protocol uses authentication groups to group users based on a common set of parameters. You should create an authentication group for iOS users. For pre-shared key and hybrid authentication, the group name must be configured on the device with the group's shared secret (pre-shared key) as the group password.

When using certificate authentication, there's no shared secret. A user's group is determined from fields in the certificate. The Cisco server settings can be used to map fields in a certificate to user groups.

RSA-Sig must be the highest priority on the ISAKMP priority list.

Certificates

When you set up and install certificates:

- The server identity certificate must contain the server's DNS name or IP address in the SubjectAltName field. The device uses this information to verify that the certificate belongs to the server. For more flexibility, you can specify the SubjectAltName using wildcard characters for per-segment matching, such as vpn.*.mycompany.com. If no SubjectAltName is specified, you can put the DNS name in the common name field.
 - The certificate of the CA that signed the server's certificate needs to be installed on the device. If it isn't a root certificate, install the rest of the trust chain so that the certificate is trusted. If you use client certificates, make sure the trusted CA certificate that signed the client's certificate is installed on the VPN server. When using certificate-based authentication, make sure the server is set up to identify the user's group, based on fields in the client certificate.
- Important:** The certificates and certificate authorities must be valid (for example, not expired). Sending of certificate chain by the server isn't supported.

IPSec settings and descriptions

IPSec has various settings that you can use to define how it will be implemented:

- *Mode:* Tunnel mode.
- *IKE Exchange Modes:* Aggressive Mode for pre-shared key and hybrid authentication or Main Mode for certificate authentication.
- *Encryption Algorithms:* 3DES, AES-128, or AES256.
- *Authentication Algorithms:* HMAC-MD5 or HMAC-SHA1.
- *Diffie-Hellman Groups:* Group 2 is required for pre-shared key and hybrid authentication. Group 2 with 3DES and AES-128 for certificate authentication. Group 2 or 5 with AES-256.
- *PFS (Perfect Forward Secrecy):* IKE phase 2, if PFS is used, the Diffie-Hellman group must be the same as was used for IKE phase 1.
- *Mode Configuration:* Must be enabled.
- *Dead Peer Detection:* Recommended.
- *Standard NAT Traversal:* Supported and can be enabled (IPSec over TCP isn't supported).
- *Load Balancing:* Supported and can be enabled.
- *Rekeying of Phase 1:* Not currently supported. It's recommended that rekeying times on the server be set to one hour.

- *ASA Address Mask*: Make sure all device address pool masks are either not set, or set to 255.255.255.255. For example:

```
asa(config-webvpn)# ip local pool vpn_users 10.0.0.1-10.0.0.254 mask 255.255.255.255.
```

If you use the recommended address mask, some routes assumed by the VPN configuration might be ignored. To avoid this, make sure your routing table contains all necessary routes and make sure the subnet addresses are accessible before deployment.

- *Application Version*: The client software version is sent to the server, letting the server accept or reject connections based on the device's software version.
- *Banner*: The banner (if configured on the server) is displayed on the device and the user must accept it or disconnect.
- *Split Tunnel*: Supported.
- *Split DNS*: Supported.
- *Default Domain*: Supported.

Per App VPN

With iOS and OS X, VPN connections can be established on a per-app basis. This provides more granular control over what data goes through VPN. With device-wide VPN, all data travels through the private network regardless of its origin. This ability to segregate traffic at the app level allows the separation of personal data and organizational data. As more and more personally owned devices are being used within organizations, Per App VPN provides secure networking for internal-use apps, while preserving the privacy of personal device activity.

Per App VPN lets each app that's managed by MDM communicate with the private network using a secure tunnel, while excluding other non-managed apps on the Apple devices from using the private network. Managed apps can be configured with different VPN connections to further safeguard data. For example, a sales quote app could use an entirely different data center than an accounts payable app, while the user's personal web browsing traffic uses the public Internet. This ability to segregate traffic at the app layer provides separation of personal data and data belonging to the organization.

In order to use Per App VPN, an app must be managed by MDM and use standard networking APIs. After enabling Per App VPN for any VPN connection, you need to associate that connection with the apps that will use it to secure the network traffic for those apps. This is done with the App-to-Per App VPN mapping payload in a configuration profile. Per App VPN is configured with an MDM configuration that specifies which apps and Safari domains are allowed to use the settings.

For information about Per App VPN support, contact third-party SSL or VPN vendors.

VPN On Demand

Overview

VPN On Demand lets Apple devices automatically establish a connection without user action. The VPN connection is started on an as-needed basis, based on rules defined in a configuration profile. VPN On Demand requires certificate-based authentication.

VPN On Demand is configured using the `OnDemandRules` key in a VPN payload of a configuration profile. Rules are applied in two stages:

- *Network Detection Stage*: Defines VPN requirements that are applied when the device's primary network connection changes.
- *Connection Evaluation Stage*: Defines VPN requirements for connection requests to domain names on an as-needed basis.

For example, rules can be used to:

- Recognize when an Apple device is connected to an internal network and VPN isn't necessary
- Recognize when an unknown Wi-Fi network is being used and require VPN for all network activity
- Require VPN when a DNS request for a specified domain name fails

Stages

VPN On Demand connects to your network in two stages.

Network detection stage

VPN On Demand rules are evaluated when the device's primary network interface changes—such as when an Apple device changes to a different Wi-Fi network, or switches to cellular on iOS or Ethernet on OS X from Wi-Fi. If the primary interface is a virtual interface, such as a VPN interface, VPN On Demand rules are ignored.

The matching rules in each set (dictionary) must all match in order for their associated action to be taken. If any one of the rules doesn't match, evaluation falls through to the next dictionary in the array, until the `OnDemandRules` array is exhausted.

The last dictionary should define a "default" configuration—that is, it should have no matching rules, only an action. This will catch all connections that haven't matched the preceding rules.

Connection evaluation stage

VPN can be triggered as needed, based on connection requests to certain domains, rather than unilaterally disconnecting or connecting VPN based on the network interface.

Rules and actions

Rules help define the type of networks associated with VPN On Demand. Actions help define what happens when matching rules are found to be true.

On Demand matching rules

Specify one or more of the following matching rules for Cisco IPSec clients:

- *InterfaceTypeMatch*: Optional. A string value of "cellular (for iOS) or Ethernet (for OS X)" or "Wi-Fi." If specified, this rule matches when the primary interface hardware is of the type specified.
- *SSIDMatch*: Optional. An array of SSIDs to match against the current network. If the network isn't a Wi-Fi network or if its SSID does not appear in the list, the match fails. Omit this key and its array to ignore SSID.
- *DNSDomainMatch*: Optional. An array of search domains as strings. If the configured DNS search domain of the current primary network is included in the array, this property matches. Wildcard prefix (*) is supported; e.g., *.example.com would match anything.example.com.
- *DNSServerAddressMatch*: Optional. An array of DNS servers addresses as strings. If all of the DNS server addresses currently configured for the primary interface are in the array, this property will match. The wildcard character (*) is supported; for example, 1.2.3.* would match any DNS servers with a 1.2.3. prefix.

- *URLStringProbe*: Optional. A server to probe for reachability. Redirection isn't supported. The URL should be to a trusted HTTPS server. The device sends a GET request to verify that the server is reachable.

Action

This required key defines VPN behavior for when all of the specified matching rules evaluate as true. Values for the Action key are:

- *Connect*: Unconditionally initiate the VPN connection on the next network connection attempt.
- *Disconnect*: Tear down the VPN connection and do not trigger any new connections on demand.
- *Ignore*: Leave any existing VPN connection up, but do not trigger any new connections on demand.
- *EvaluateConnection*: Evaluate the ActionParameters for each connection attempt. When this is used, the key ActionParameters, described below, is required to specify the evaluation rules.
- *Allow*: For iOS devices with iOS 6 or earlier, see [Backward compatibility](#).

ActionParameters

This is an array of dictionaries with the keys described below, evaluated in the order in which they occur. Required when Action is EvaluateConnection.

- *Domains*: Required. An array of strings that define the domains for which this evaluation applies. Wildcard prefixes are supported, such as *.example.com.
- *DomainAction*: Required. Defines VPN behavior for the domains. Values for the DomainAction key are:
 - *ConnectIfNeeded*: Brings up VPN if DNS resolution for the domains fails, such as when the DNS server indicates it can't resolve the domain name, or if the DNS response is redirected, or if the connection fails or times out.
 - *NeverConnect*: Don't trigger VPN for the domains.

When DomainAction is ConnectIfNeeded, you can also specify the following keys in the connection evaluation dictionary:

- *RequiredDNSServers*: Optional. An array of IP addresses of DNS servers to be used for resolving the domains. These servers don't need to be part of the device's current network configuration. If these DNS servers aren't reachable, VPN will be triggered. For consistent connections, configure an internal DNS server or a trusted external DNS server.
- *RequiredURLStringProbe*: Optional. An HTTP or HTTPS (preferred) URL to probe, using a GET request. If DNS resolution for this server succeeds, the probe must also succeed. If the probe fails, VPN is triggered.

Backward compatibility

Before iOS 7, domain triggering rules were configured from arrays of domains:

- OnDemandMatchDomainAlways
- OnDemandMatchDomainOnRetry
- OnDemandMatchDomainNever

The OnRetry and Never cases are still supported in iOS 7 or later, although deprecated in favor of the EvaluateConnection action.

To create a profile that works on both iOS 7 and earlier releases, use the new `EvaluateConnection` keys in addition to the `OnDemandMatchDomain` arrays. Earlier versions of iOS that don't recognize `EvaluateConnection` use the old arrays; iOS 7 or later uses `EvaluateConnection`.

Old configuration profiles that specify the `Allow` action should work on iOS 7 or later, with the exception of `OnDemandMatchDomainsAlways` domains.

Always-on VPN

Overview

Always-on VPN gives your organization full control over device traffic by tunneling all IP traffic back to the organization. The default tunneling protocol, IKEv2, secures traffic transmission with data encryption. Your organizations can now monitor and filter traffic to and from its devices, secure data within its network, and restrict device access to the Internet.

Always-on VPN activation requires device supervision. Once the Always-on VPN profile is installed on a device, Always-on VPN automatically activates with no user interaction. Always-on VPN stays activated (including across reboots) until the Always-on VPN profile is uninstalled.

With Always-on VPN activated on the device, the VPN tunnel bring-up and teardown is tied to the interface IP state. When the interface gains IP network reachability, tunnel establishment is attempted. When the interface IP state goes down, the tunnel is torn down. Always-on VPN also supports per-interface tunnels. For iOS devices, there'll be one tunnel for each active IP interface (that is, one tunnel for the cellular interface, and one tunnel for the Wi-Fi interface). As long as the VPN tunnel or tunnels are up, all IP traffic is tunneled. All traffic includes all IP-routed traffic and all IP-scoped traffic (that is, traffic from first-party apps such as FaceTime and Messages). If the tunnel or tunnels aren't up, all IP traffic is dropped.

All traffic tunneled from a device will reach a VPN server. You can apply optional filtering and/or monitoring treatments before forwarding the traffic to its destination within your organization's network or the Internet. Similarly, traffic to the device will be routed to your organization's VPN server, where filtering and/or monitoring treatments may be applied before being forwarded to the device.

Deployment scenarios

iOS devices runs in single-user mode. There's no distinction between device identity and user identity. When an iOS device establishes a IKEv2 tunnel to the IKEv2 server, the server perceives the iOS device as a single peer entity. Traditionally, there is one tunnel between a pair of iOS devices and a VPN server. Since Always-on VPN introduces per-interface tunnels, there may be multiple simultaneous tunnels established between a single iOS device and the IKEv2 server, depending on the deployment model.

Always-on VPN configuration supports the following deployment models, fulfilling different solution requirements.

Cellular-only devices

If your organization chooses to deploy Always-on VPN on cellular-only iOS devices (Wi-Fi interface permanently taken out or deactivated), one IKEv2 tunnel is established over the cellular IP interface between each device and the IKEv2 server. This is the same as the traditional VPN model. The iOS device acts as one IKEv2 client, with one identity (i.e. one client certificate or one user and password) establishing one IKEv2 tunnel with the IKEv2 server.

Cellular and Wi-Fi devices

If your organization chooses to deploy Always-on VPN for iOS devices with both cellular and Wi-Fi interfaces, two simultaneous IKEv2 tunnels will be established from the device. There are two scenarios using cellular and Wi-Fi devices:

- Cellular tunnel and Wi-Fi tunnel terminating on separate IKEv2 servers

Always-on VPN per-interface tunnel configuration keys allow your organization to configure devices establishing a cellular tunnel to one IKEv2 server and Wi-Fi tunnel to a second IKEv2 server. One benefit of this model is that a device can use the same client identity (that is, client certificate or user/password) for both tunnels since the tunnels terminate on different servers. With different servers, your organization also has greater flexibility on per-interface-type traffic (cellular traffic vs Wi-Fi traffic) segregation and control. The drawback is that your organization has to maintain two different IKEv2 servers with identical client authentication policies.

- Cellular tunnel and Wi-Fi tunnel terminating on same IKEv2 servers

Always-on VPN per-interface tunnel configuration also lets your organization configure a device to establish the cellular tunnel and the Wi-Fi tunnel to the same IKEv2 server.

Client identity usage:

- *One client identity per device:* Your organization can configure the same client identity (that is, one client certificate or one user/password pair) for both a cellular tunnel and Wi-Fi tunnel, if the IKEv2 server supports multiple tunnels per client. The benefit is that you can avoid the extra client identity per device and the extra configuration/resource burden on the server. The drawback is that as a device moves in and out of networks, new tunnels get established and old tunnels become stale. Depending on the server implementation, the server may not be able to clean up stale tunnels efficiently and accurately. Your organization must implement a strategy for stale tunnel cleanup on the server.
- *Two client identities per device:* Your organization can configure two client identities (that is, two client certificates or two user/password pairs), one for a cellular tunnel and one for a Wi-Fi tunnel. The IKEv2 server sees two different clients establishing their own tunnel. The benefit of this model is that it works with most server implementations, since many servers differentiate tunnels by their client identities and only allow one tunnel per client. The drawback of this model is doubled client identity management and doubled configuration and resource management on the server.

Always-on VPN configuration profile

An Always-on VPN configuration profile can be composed either manually, using one of the Apple configuration profile editors such as Profile Manager, Apple Configurator, or a third-party MDM vendor. For more information, see [Profile Manager Help](#) or [Apple Configurator Help](#).

User interaction keys

To keep users from deactivating the Always-on VPN feature, disallow removal of the Always-on VPN profile by setting the top-level profile key “PayloadRemovalDisallowed” to true.

To keep users from altering Always-on VPN feature behavior by installing other configuration profiles, disallow UI profile installation by setting the allowUIConfigurationProfileInstallation key to false under the com.apple.applicationaccess payload. Your organization can implement additional restrictions using other supported keys under the same payload.

Certificate payloads

- *Server CA Certificate:* If the IKEv2 tunnel authentication method is to use certificates, the IKEv2 server sends its server certificate to the iOS device, which validates server identity. In order for the iOS device to validate the server certificate, it needs the server's Certificate Authority (the issuer of the server certificate) certificate. The server CA certificate may have already been installed onto the device previously. Otherwise, your organization can include the server CA certificate by creating a certificate payload for the server CA certificate.
- *Client CA Certificate(s):* If the IKEv2 tunnel authentication method is to use certificates or EAP-TLS, the iOS device sends its client certificates to the IKEv2 server, which validates the client identity. The client may have one or two client certificates, depending on the deployment model selected. Your organization needs to include the client certificate(s) by creating certificate payload(s) for the client certificate(s). At the same time, for the IKEv2 server to validate the client identity, the IKEv2 server needs to have the client's Certificate Authority (the issuer of the client certificates) certificate installed.
- *Always-on VPN IKEv2 Certificate Support:* Currently, Always-on VPN IKEv2 supports only RSA certificates.

Always-on VPN payload

The following apply to the Always-on VPN payload.

- The Always-on VPN payload can be installed only on supervised iOS devices
- A configuration profile can contain only one Always-on VPN payload
- Only one Always-on VPN configuration profile can be installed on an iOS device at a time

Connect Automatically in iOS

Always-on VPN provides an optional "UIToggleEnabled" key to let your organization enable a "Connect Automatically" toggle in the VPN Settings. If this key isn't specified in the profile or is set to 0, Always-on VPN attempts to bring up one or two VPN tunnels. If this key is set to 1, the toggle is presented in the VPN Settings pane and the user has the choice to turn on/off VPN tunneling. If the user chooses to turn off VPN tunneling, no tunnel is established and the device drops all IP traffic. This is useful in the case when there's no IP reachability and the user still wants to make phone calls. The user can turn off VPN tunneling to avoid unnecessary attempts to bring up a VPN tunnel.

Per-interface tunnel configuration array

At least one tunnel configuration is required (that is, applied to the cellular interface for cellular-only devices, or applied to both cellular and Wi-Fi interfaces) in the TunnelConfigurations array. At most, two tunnel configurations can be included (one for cellular interfaces and one for Wi-Fi interfaces).

Captive Traffic Exceptions

Always-on VPN only supports Captive AutoLogon (automatic logging on to supported Captive networks with pre-assigned credentials, such as credentials derived from SIM).

Always-on VPN also provides control over Captive handling by supporting the following:

- *AllowCaptiveWebSheet:* A key to allow traffic from built-in Captive WebSheet App to pass outside the tunnel. WebSheet App is a browser that handles Captive logon if no third-party Captive App is present. Your organization should consider the security risk of using this key, because the WebSheet is a functional browser capable of rendering any content from the responding Captive server. Allowing traffic for WebSheet makes the device vulnerable to misbehaving or malicious Captive servers.

- *AllowAllCaptiveNetworkPlugins*: A key to allow traffic from all entitled third-party Captive Apps outside the tunnel. This key takes precedence over the *AllowedCaptiveNetworkPlugins* dictionary.
- *AllowedCaptiveNetworkPlugins*: A list of bundle IDs of entitled third-party Captive Apps. Traffic from this list of third-party Captive Apps is allowed outside the tunnel. If the *AllowAllCaptiveNetworkPlugins* key is also configured, this list won't take effect.

Service exceptions

Always-on VPN tunnels all IP traffic by default. This includes all local traffic and traffic for cellular carrier services. So the default Always-on VPN behavior doesn't support any local IP service or IP carrier service. Always-on VPN Service Exceptions support lets your organization alter the default treatment of service traffic to either pass outside the tunnel or drop. The currently supported services are VoiceMail and AirPrint, and the allowed action is either Allow (to allow outside the tunnel) or Drop (to drop regardless of the tunnel).

For more information about Always-on VPN IKEv2 protocol keys and attributes, see [Configuration Profile Key Reference](#) at the iOS Developer Library website.

Internet services

5

Overview

Internet services from Apple have been built with the same security goals that iOS promotes throughout the platform—secure handling of data, whether at rest on the iOS device or in transit over wireless networks, protection of users’ personal information, and threat protection against malicious or unauthorized access to information and services. Each service uses its own powerful security architecture without compromising the overall ease of use of iOS.

These services help users communicate and create and backup their personal data, all without compromising your organization’s data.

They include:

- Apple ID
- Find My iPhone and Activation Lock
- Continuity
- iCloud
- iCloud Keychain
- iMessage
- FaceTime
- Siri
- Apple ID for Students

You can use an MDM solution and iOS restrictions to restrict particular services. For information, see the MDM restrictions [Overview](#).

At Apple, security and privacy are fundamental to the design of all our hardware, software, as well as our services. That’s why we respect our customers’ privacy and protect it with strong encryption, plus strict policies that govern how all data is handled. For more information, see www.apple.com/privacy.

Apple ID

An Apple ID is required for anyone who wants to access services from Apple. You need to understand Apple IDs, so you can educate your users about how to set up their own.

An Apple ID is an identity that’s used to log in to various Apple services such as FaceTime, iMessage, the iTunes Store, App Store, iBooks Store, and iCloud. These services give users access to a wide range of content to streamline business tasks, increase productivity, and support collaboration.

To get the most out of these services, users should have their own Apple ID. If they don't have one, they can create one even before they receive an Apple device or use Setup Assistant. Using Setup Assistant gives the user an easy and streamlined way to create an Apple ID right from their Apple device. Apple IDs can also be created without the need for a credit card.

For one-to-one and student-owned deployments (or BYOD deployments), each user should have their own Apple ID. In a shared-use deployment, an institution-owned Apple ID can be used to deploy content on multiple Apple devices.

With an Apple ID, each student or employee can install apps, books, and other content provided by the institution; take notes in iBooks that can be accessed between iOS devices and Mac computers; and enroll in iTunes U courses—all without the need for IT to manage the Apple ID on the user's Apple device.

For more information about Apple IDs, see the [My Apple ID](#) website.

Find My iPhone and Activation Lock

If an iOS device is lost or stolen, it's important to deactivate and erase it. With Find My iPhone, part of the iCloud suite, users can see the last reported location of their iPad, iPhone, or iPod touch by using Find My iPhone on iCloud.com or the Find My iPhone app on an iOS device. Once the iOS device is located, the user can play a sound on it, put it in Lost Mode, or erase it completely if it's connected to the Internet.

Lost Mode (iOS 6 or later) will lock the iOS device with a passcode, display a custom message on the screen, and keep track of its location. For iOS devices with iOS 5, this feature will only lock the device.

With iOS 7 or later, when Find My iPhone is turned on, the iOS device can't be reactivated without entering the owner's Apple ID credentials. You should supervise your organization's devices and have a policy in place for users to disable the feature so that Find My iPhone doesn't prevent your organization from assigning the device to another individual.

With iOS 7.1 or later, you can use a compatible MDM solution to enable Activation Lock on supervised devices when a user turns on Find My iPhone. MDM administrators can manage Find My iPhone Activation Lock by supervising devices with Apple Configurator or the Device Enrollment Program. Your MDM solution can then store a bypass code when Activation Lock is enabled, and later use this code to clear Activation Lock automatically when you need to erase the device and assign it to a new user. See your MDM solution documentation for details.

Important: By default, supervised devices never have Activation Lock enabled, even if the user turns on Find My iPhone. However, an MDM server may retrieve a Bypass Code and permit Activation Lock on the device. If Find My iPhone is turned on when the MDM server enables activation lock, activation lock is enabled at that point. If Find My iPhone is turned off when the MDM server enables activation lock, it's enabled the next time the user activates Find My iPhone.

For more information about Find My iPhone, Lost Mode, and Activation Lock, see the Apple Support articles [iCloud Support](#), [iCloud: Use Lost Mode](#), and [Mobile Device Management and Find My iPhone Activation Lock](#). Also see [Activation Lock settings](#) in Profile Manager Help.

Continuity

Continuity is a suite of features that allow a Mac and iPhone or iPad to communicate together seamlessly. Continuity requires iOS 8 or later and OS X Yosemite or later, and it may require the devices to be registered with the same Apple ID.

Note: Some features may not be available in all countries, regions, or all languages.

Phone calls

An iPhone and Mac work seamlessly together when making or answering phone calls. If a user is working on their Mac and their iPhone is nearby, they can make or answer a call on their Mac and, if they choose, pick up their iPhone and continue the call.

SMS

Users can communicate using SMS from their iOS device with iOS 8.1 or later, and from OS X Yosemite or later. SMS messages appear on all the user's devices, letting them respond on any device.

Handoff

A user can begin writing a message in Mail or create a Pages document on their Mac, and when they transition to their nearby iOS device with iOS 8 or later, the item they were working on is already there, ready for continued editing. A user will see a small icon in the corner of the iOS device or in the Dock on the Mac. Swiping on an iOS device or clicking on the Mac brings up the document. Handoff works with Calendar, Contacts, Mail, Maps, Messages, Pages, Numbers, Keynote, Reminders, and Safari. App developers can also build Handoff into their apps.

Instant Hotspot

Instant Hotspot lets a Mac use an iPhone or iPad (with cellular connectivity) with iOS 8.1 or later as an Internet connection when Wi-Fi isn't available. The signal strength and battery life of your iOS device shows in the Menu bar of the Mac. Once the user disconnects from the iOS device, the hotspot deactivates to preserve iOS battery life.

Note: Check with the carrier for hotspot availability.

AirDrop

AirDrop lets a Mac with OS X Mavericks or later and a nearby iOS device with iOS 8 or later wirelessly share files without a wireless network present. AirDrop works from any sharing menu and in the Finder sidebar on a Mac.

iCloud

iCloud lets users store personal content such as contacts, calendars, documents, and photos, and keep them up to date across multiple iOS devices and Mac computers. iCloud secures your content by encrypting it when sent over the Internet, storing it in an encrypted format and using secure tokens for authentication. iOS devices use iCloud Backup to back up information—including iOS device settings, app data, and text and MMS messages—daily over Wi-Fi. iCloud Backup works only when the device is locked, is connected to a power source, and has Wi-Fi access to the Internet. And iCloud offers the ability to locate lost or stolen iOS devices or Mac computers using Find my iPhone.

An MDM solution can also keep managed apps from being backed up to iCloud. This gives users the benefits of using iCloud for personal data while keeping corporate information from being stored in iCloud. Data from corporate accounts and enterprise in-house apps isn't backed up to iCloud. Some services—such as iCloud Photos, iCloud Keychain, and iCloud Drive—can be disabled through restrictions entered manually on the device or set by configuration profiles.

For more information about iCloud, see the [iCloud](#) website. For more information about iCloud security and privacy, see the Apple Support article [iCloud security and privacy overview](#). For more information about system requirements for iCloud, see the Apple Support article [System requirements for iCloud](#).

Note: Some features require a Wi-Fi connection. Some features aren't available in all countries. Access to some services is limited to 10 devices.

iCloud Drive

Users can safely store their documents on iCloud Drive and access them anywhere, anytime from their iPhone, iPad, Mac, or Windows PC. iOS app document libraries are also accessible from a Mac, so a document started on an iOS device can be edited on a Mac.

Users can also share their Pages, Numbers, and Keynote documents stored in iCloud Drive with others. Each iOS app shows compatible documents stored in iCloud Drive. On a Mac, iCloud Drive appears as a folder in OS X. Users drag and drop to add files, organize with folders and tags, and even search using Spotlight.

iCloud keeps your information up to date on all your devices. Any changes made to a file while offline are automatically updated as soon as the device is back online.

iCloud Keychain

iCloud Keychain specifically keeps website passwords used in Safari and Wi-Fi network passwords up to date on all your iOS devices and Mac computers set up for iCloud. It can store passwords for other apps that support it. iCloud Keychain also stores credit card information you save in Safari, so Safari can autofill it on your iOS devices and Mac. iCloud Keychain also stores Internet account sign-in and configuration information.

iCloud Keychain consists of two services:

- Keeping Keychain up-to-date on all devices
- Keychain recovery

Keeping the Keychain up-to-date on iOS devices and Mac computers requires devices to participate only after user approval, and each keychain item that's eligible is exchanged with per-device encryption via iCloud key value storage. The keychain items are temporary and don't persist in iCloud after being synced.

Keychain recovery lets users save their keychain with Apple, without giving Apple the ability to read the passwords and other data it contains. Even if the user has only a single iOS device or Mac, keychain recovery provides a safety net against data loss. This is particularly important when Safari is used to generate random, strong passwords for web accounts, because the only record of those passwords is in the keychain.

The user's iCloud Keychain is backed up in iCloud if the user creates an iCloud Security Code. Secondary authentication and a secure escrow service are important features of keychain recovery. The user's keychain is encrypted using a strong passcode, and the escrow service provides a copy of the keychain only if a strict set of conditions is met.

Important: If the user doesn't create an iCloud Security Code, Apple can't help recover the iCloud Keychain. See the Apple Support article [Frequently asked questions about iCloud Keychain](#).

iMessage

iMessage is a messaging service for both iOS devices and Mac computers that enables one-to-one or group chats. iMessage supports text and attachments such as photos, contacts, and locations. Messages appear on all of a user's registered iOS devices and Mac computers, so the user can continue a conversation on any one of them. iMessage uses the Apple Push Notification Service (APNs) and end-to-end encryption with keys known only to the sending and receiving iOS devices and Mac computers. Apple can't decrypt messages, and messages aren't logged.

Note: Normal carrier data rates may apply. Messages may be sent as SMS when iMessage is unavailable; carrier messaging fees apply.

FaceTime

FaceTime is Apple's video and audio calling service. FaceTime calls use the Apple Push Notification Service to establish a connection, then use Internet Connectivity Establishment (ICE) and Session Initiation Protocol (SIP) to create an encrypted stream. Users can communicate between any mix of iOS and OS X devices using FaceTime.

Note: FaceTime calling requires a FaceTime-enabled device for both the caller and recipient, and a Wi-Fi connection. FaceTime over a cellular network requires iPhone 4s or later, iPad with Retina Display or later, or iPad mini or later with cellular data capability. Availability over a cellular network depends on carrier policies. Data charges may apply.

Siri

By simply talking naturally, users can have Siri send messages, schedule meetings, place phone calls, and more. Siri uses speech recognition, text-to-speech, and a client-server model to respond to a broad range of requests. The tasks that Siri supports have been designed to ensure that only the absolute minimum of personal information is used, and that the information used is fully protected. Siri requests and voice recordings aren't personally identified, and whenever possible, Siri functions are carried out on the iOS device, not the server.

Note: Siri may not be available in all languages or in all areas, and features may vary by area. Internet access is required. Cellular data charges may apply.

Apple ID for Students

The Apple ID for Students program is for students under the age of 13. Apple IDs are requested by the school or school district, and created by Apple after receiving from a parent or guardian a signed Parent Privacy Disclosure and Consent form. This method complies with the Children's Online Privacy Protection Act (COPPA).

For more information about Apple ID for Students, see:

- [Apple ID for Students website](#)
- [Apple ID for Students Help](#)

Note: The Apple ID for Students program isn't available in all countries or regions.

Apple Push Notification Service (APNs)

Many services rely on Apple Push Notification Service (APNs). APNs is a key part of how Apple devices learn of updates, MDM policies, and incoming messages. In order for your Apple devices to work with these services, you need to allow network traffic from the device to Apple's network (17.0.0.0/8) on port 5223, with a fallback option of port 443.

This traffic is a secured, binary protocol specific to APNs, and can't go through a proxy. Attempts to inspect the traffic or reroute it will result in the client, APNs, and push provider servers marking the network conversation as compromised and invalid.

There are multiple layers of security applied to APNs at the endpoints and the servers.

To read technical information about these precautions, see [Local and Remote Notification Programming Guide](#).

Overview

iOS and OS X are built with multiple layers of security, so Apple devices can securely access network services and protect important data. iOS and OS X also provide secure protection through the use of passcode and password policies that can be delivered and enforced with MDM. And if an Apple device falls into the wrong hands, a user or IT administrator can use a remote command to erase all private information.

Ensuring the security of Apple devices for enterprise use involves the following:

- Methods that prevent unauthorized use of the device
- Protecting data at rest, even when the device is lost or stolen
- Networking protocols and the encryption of data in transmission
- Enabling apps to run securely and without compromising platform integrity

These capabilities work together to provide a secure mobile computing platform. To learn more about security with iOS, see [iOS and the new IT](#).

Device and data security

Overview

Establishing strong policies for access to Apple devices is critical to protecting your organization's information. Strong iOS device passcodes are the frontline of defense against unauthorized access, and they can be configured and enforced with MDM.

iOS devices use the unique passcode established by each user to generate a strong encryption key that's used to further protect mail and sensitive app data on the device. iOS also provides secure methods to configure devices in an IT environment, where specific settings, policies, and restrictions must be in place. These methods provide flexible options for establishing a standard level of protection for authorized users.

Passcode policies

An iOS device passcode keeps unauthorized users from accessing data on the device. iOS lets you choose from an extensive set of passcode policies to meet your security needs.

These passcode policies include:

- Require passcode on iOS device
- Require alphanumeric value
- Minimum passcode length
- Minimum number of complex characters
- Maximum passcode age
- Time before auto-lock

- Passcode history
- Grace period for device lock
- Maximum number of failed attempts before the iOS device will be erased

Policy enforcement

You can distribute policies in a configuration profile that users install. You can also define a profile so that deleting the profile is possible only with an administrator password, or you can define the profile so that it's locked to the iOS device and can't be removed without completely erasing all of the device's contents. Passcode settings configured remotely with MDM can push policies directly to the device, letting policies be enforced and updated without any action by the user.

If a device is configured to access a Microsoft Exchange account, Exchange ActiveSync policies are pushed to the device wirelessly. The available set of policies varies, depending on the version of Exchange ActiveSync and Exchange Server. If both Exchange and MDM policies exist, the more stringent policy is applied.

Secure device configuration

A configuration profile is an XML file that contains device security policies and restrictions, VPN configuration information, Wi-Fi settings, mail and calendar accounts, and authentication credentials that let iOS devices work with your IT systems. The ability to establish passcode policies along with device settings in a configuration profile ensures that devices are configured correctly and according to security standards set by your IT department. Because configuration profiles can be encrypted and locked, the settings can't be removed, altered, or shared with others.

Configuration profiles can be both signed and encrypted. Signing a configuration profile ensures that the settings it enforces cannot be altered in any way. Encrypting a configuration profile protects the profile's contents and permits installation only on the device it was created for. Configuration profiles are encrypted using CMS (Cryptographic Message Syntax, [RFC 3852](#)), supporting 3DES and AES 128.

The first time you distribute an encrypted configuration profile, you can install it with a USB connection using Apple Configurator, wirelessly using the Over-the-Air Profile Delivery and Configuration protocol, or via MDM. Subsequent encrypted configuration profiles can be delivered by a mail message attachment, hosted on a website accessible to your users, or pushed to the device with MDM.

For more information, see [Over-the-Air Profile Delivery and Configuration](#).

Data protection

You can make sensitive data such as mail messages and attachments stored on the device more secure by using data protection features built into iOS. Data protection uses each user's unique device passcode, along with the hardware encryption on iOS devices, to generate a strong encryption key. This prevents data from being accessed when the device is locked, and ensures that critical information is secured even if the device is compromised.

To turn on data protection, establish a passcode on the device. The effectiveness of data protection depends on a strong passcode, so it's important to require a passcode stronger than four digits.

Users can make sure that data protection is enabled on their device by looking at the passcode settings screen. Mobile device management solutions are able to query the device for this information as well.

There are also data protection APIs for developers, which can be used to secure data within App Store apps or custom-developed in-house apps. With iOS 7 or later, data stored by apps is, by default, in the security class “Protected Until First User Authentication.” This is similar to full-disk encryption on desktop computers, and protects data from attacks that involve a reboot.

iOS 8 includes data protection for Calendars, Contacts, Messages, Notes, Reminders, and managed books and PDFs.

Note: If a device is upgraded from iOS 6, existing data stores aren’t converted to the new class. Removing and reinstalling the app causes the app to receive the new protection class.

Encryption

iOS devices use hardware encryption. Hardware encryption uses 256-bit AES to protect all data on the device. Encryption is always enabled, and cannot be disabled. Additionally, data backed up in iTunes to a user’s computer can be encrypted. This can be enabled by the user, or enforced by using device restriction settings in configuration profiles.

The cryptographic modules in iOS 6 or later have been validated to comply with U.S. Federal Information Processing Standard (FIPS) 140-2 Level 1. This validates the integrity of cryptographic operations in Apple apps and third-party apps that properly use iOS cryptographic services.

For more information, see the Apple Support articles [iOS product security: Validations and guidance](#) and [Apple FIPS iOS Cryptographic Modules v4.0](#).

Per-message S/MIME

iOS 8 and OS X Yosemite support per-message S/MIME, so S/MIME users can choose to always sign and encrypt by default or selectively sign and/or encrypt individual messages for greater control over the security of each mail message.

Certificates for use with S/MIME can be delivered to the Apple device using a configuration profile, MDM, or SCEP. This gives IT the flexibility needed to ensure that users always have the appropriate certificates installed.

External email addresses

iOS 8 and OS X Yosemite support creating a domain list of specific suffixes. Mail messages that aren’t addressed to domains in the approved list are marked in red. For example, a user could have both example.com and group.example.com in their list of known domains. If a user with example.com and group.example.com in their known domains list were to enter anyone@acme.com in a Mail message, that address would be clearly marked so the user would know the domain acme.com wasn’t on their approved list.

Touch ID

Touch ID is the fingerprint-sensing system built into some iOS devices, making highly secure access to the device faster, easier, and more secure. This technology reads fingerprints in any orientation and learns more about the user’s fingerprint over time, with the sensor continuing to expand the fingerprint map as additional overlapping nodes are identified with each use.

Touch ID makes using a longer, more complex passcode more practical, because the user doesn’t have to enter it as often.

When Touch ID is enabled, the device immediately locks when the Sleep/Wake button is pressed. With passcode-only security, many users set an unlocking grace period to avoid having to enter a passcode each time they use the device. With Touch ID, the device locks every time it goes to sleep, and requires a fingerprint—or optionally, the passcode—on waking.

Touch ID works with the Secure Enclave, a coprocessor in the Apple A7 chip. The Secure Enclave has its own protected, encrypted memory space and communicates securely with the Touch ID sensor. When the device locks, the keys for Data Protection Class Complete are protected by a key kept in the encrypted memory of the Secure Enclave. The key is held for a maximum of 48 hours, and is discarded if the device is rebooted or an unrecognized fingerprint is used five times. If a fingerprint is recognized, the Secure Enclave provides the key for unwrapping the Data Protection keys and the device is unlocked.

iOS 8 introduces the use of Touch ID to sign in to third-party apps. If the developer has integrated this capability into their app, there is no need for the user to enter a password. Any keychain item specified by the developer can be unlocked using Touch ID. A user's fingerprint data is protected and never accessed by iOS or by apps.

Remote wipe

Apple devices fully support remote wipe. If an Apple device is lost or stolen, an administrator or the owner of the device can issue a remote wipe command that removes all data and deactivates the device using an MDM solution or the Find My iPhone feature of iCloud. If the device is configured with an Exchange account, the administrator can initiate a remote wipe command using the Exchange Management Console (Exchange Server 2007) or Exchange ActiveSync Mobile Administration Web Tool (Exchange Server 2003 or 2007). Users of Exchange Server 2007 can initiate a remote wipe command directly, using Outlook Web Access.

Local wipe

You can configure devices to automatically initiate a local wipe after several failed passcode attempts. This protects against brute-force attempts to gain access to the device. When a passcode is established, users can turn on local wipe directly within settings. By default, iOS automatically wipes the device after 10 failed passcode attempts. The maximum number of failed attempts can be set in a configuration profile, set by an MDM server, or enforced over the air by Microsoft Exchange ActiveSync policies.

Network security

Mobile users must be able to access corporate networks from anywhere in the world, yet it's also important to ensure that users are authorized and that their data is protected during transmission. Built-in network security technologies in iOS accomplishes these security objectives for both Wi-Fi and cellular connections.

iOS network security supports:

- Built-in Cisco IPSec, L2TP, IKEv2, PPTP
- SSL VPN via App Store apps
- SSL/TLS with X.509 certificates
- WPA/WPA2 Enterprise with 802.1X
- Certificate-based authentication
- RSA SecurID, CRYPTOCARD

VPN

Many enterprise environments have some form of virtual private network (VPN). These secure network services typically require minimal setup and configuration to work with Apple devices, which integrate with a broad range of commonly used VPN technologies.

For details, see the Virtual Private Networks (VPN) [Overview](#).

IPSec

iOS and OS X support IPSec protocols and authentication methods. For details, see [Supported protocols and authentication methods](#).

SSL/TLS

iOS supports SSL v3 and Transport Layer Security (TLS v1.0, 1.1, and 1.2). Safari, Calendar, Mail, and other Internet apps automatically use these to enable an encrypted communication channel between iOS and OS X and corporate services.

WPA/WPA2

iOS and OS X support WPA2 Enterprise to provide authenticated access to your enterprise wireless network. WPA2 Enterprise uses 128-bit AES encryption, so user data is protected when communicating over a Wi-Fi network connection. And with support for 802.1X, the Apple devices can be integrated into a broad range of RADIUS authentication environments.

iOS and OS X support these 802.1X authentication protocols:

- EAP-TLS
- EAP-TTLS
- EAP-FAST
- EAP-SIM
- EAP-AKA
- PEAP v0, v1
- LEAP

For more information, see the Wi-Fi [Overview](#).

FaceTime and iMessage encryption

Each FaceTime session and iMessage conversation is encrypted. iOS and OS X create a unique ID for each user, ensuring communications are encrypted, routed, and connected properly.

App security

To ensure apps can't be tampered with, iOS and OS X include a sandboxed approach to app runtime protection and app signing. iOS and OS X also have the Keychain, a framework which facilitates secure storage of app and network service credentials in an encrypted storage location. For iOS and OS X developers, it offers a Common Crypto architecture that can be used to encrypt data that apps store.

Runtime protection

All apps from the App Store are sandboxed to restrict access to data stored by other apps. Also, system files, resources, and the kernel are shielded from the user's app space. If an app needs to access data from another app, it can do so only by using the APIs and services provided by iOS and OS X. Code generation is also prevented.

Mandatory code signing

All apps from the App Store must be signed. The apps provided with Apple devices are signed by Apple. Third-party apps are signed by the developer, using an Apple-issued certificate. This ensures that apps haven't been tampered with or altered. Runtime checks are made to ensure that an app hasn't become untrusted since it was last used.

You can control the use of custom in-house apps with a provisioning profile. Users must have the provisioning profile installed to start the app. Provisioning profiles can be installed over the air via MDM. You can also restrict the use of an app to specific devices.

Secure authentication framework

iOS and OS X provide a secure, encrypted keychain for storing digital identities, user names, and passwords. Keychain data is partitioned and protected with Access Control Lists (ACLs), so credentials stored by third-party apps can't be accessed by apps with a different identity unless the user explicitly approves them. This provides the mechanism for securing authentication credentials on Apple devices across a range of apps and services within your organization.

Common Crypto architecture

App developers can use encryption APIs to protect their app data. Data can be symmetrically encrypted using proven methods such as AES, RC4, or 3DES. iOS devices and current Intel Mac computers also provide hardware acceleration for AES encryption and SHA1 hashing, maximizing app performance.

App data protection

Apps can also take advantage of the built-in hardware encryption on iOS devices to further protect sensitive app data. Developers can designate specific files for data protection, instructing the system to make the contents of the file cryptographically inaccessible to both the app and any potential intruders when the device is locked.

App entitlements

By default, an iOS device app has very limited privileges. Developers must explicitly add entitlements to use most features, such as iCloud, background processing, or shared keychains. This ensures that apps can't grant themselves data access they weren't deployed with. iOS apps must ask for explicit user permission before using many iOS features, such as GPS location, user contacts, the camera, and stored photos.

Single Sign-On and Touch ID

Developers can take advantage of Single Sign-On and Touch ID to provide secure, seamless authentication integration between different apps and permit authentication using Touch ID.

For more information, see [Configure Single Sign-On](#) and [Touch ID](#).

Configuration and management

7

Overview

You can streamline Apple device deployments through management techniques that simplify account setup, configure institutional policies, distribute apps, and apply restrictions. You can configure iOS and OS X preferences and accounts manually, or with an MDM solution. Users can then do most of the initial setup themselves through the Setup Assistant built into the Apple devices. And after the devices are configured and enrolled in MDM, they can be managed wirelessly by your IT department.

MDM gives your organization the ability to securely enroll Apple devices in the corporate or educational environment, wirelessly configure and update settings, monitor policy compliance, deploy apps, and remotely wipe or lock them if they're managed. Several MDM solutions are available for different server platforms. Each solution offers its own management console, features, and pricing. Before you choose an MDM solution, review this section to see which features are most important to your organization.

Depending on who owns the Apple devices and how they're deployed, several different configuration workflows and capabilities are possible. For more information, see the Deployment models [Overview](#).

This section describes the complete set of tools, programs, and services available to support your Apple device deployment.

Setup Assistant and activation

iOS and OS X provide Setup Assistant to activate each new or erased Apple device, configure basic settings, and personalize preferences such as language, location services, Siri, iCloud, and Find My iPhone. Users can take an Apple device right out of the box and use these features to get up and running, or your organization can perform these basic setup tasks. Setup Assistant also let users create a personal Apple ID, if they don't have one already.

Apple devices enrolled in the Device Enrollment Program and managed by MDM can have these Setup Assistant screens skipped:

- *Restore from backup*: Doesn't restore from backup
- *Apple ID*: Doesn't prompt the user to sign in with an Apple ID
- *Terms and Conditions*: Skips the Terms and Conditions
- *Send diagnostics*: Doesn't automatically send diagnostic information
- *Location (iOS only)*: Doesn't enable Location Services
- *Touch ID (iOS only)*: Doesn't enable Touch ID
- *Passcode (iOS only)*: Skips the passcode setup
- *Apple Pay (iOS only)*: Doesn't enable Apple Pay
- *Siri (iOS only)*: Doesn't enable Siri

- *Display Zoom (iOS only)*: Doesn't enable Display Zoom
- *Registration (OS X only)*: Doesn't permit registration
- *FileVault (OS X only)*: Doesn't enable FileVault

Unless these items are also permanently restricted using the MDM solution, users can perform any of these after the Apple device is set up.

For more information about the Device Enrollment Program see:

- [Device Enrollment Program](#)
- [Device Enrollment Program](#)
- [Apple Deployment Programs Help](#)

Configuration profiles

A configuration profile is an XML file you use to distribute configuration information to Apple devices. Configuration profiles automate the configuration of settings, accounts, restrictions, and credentials. They can be installed through a mail message attachment, downloaded from a webpage, or installed on iOS devices with Apple Configurator. If you need to configure a large number of iOS devices, or just prefer a hands-off over-the-air deployment model, you can deliver configuration profiles through MDM.

Configuration profiles that contain certificate and Wi-Fi payloads can also be installed on Apple TV. For more information, see the Apple Support article [How to install a configuration profile on Apple TV](#).

Configuration profiles can be encrypted and signed, which lets you restrict their use to a specific Apple device and prevents anyone from changing the settings that the profile contains. An MDM administrator can also mark a profile as being locked to the device, so once installed, it can be removed only by wiping the device of all data or, optionally, by entering a password.

With the exception of passcodes, users can't change the settings provided in a configuration profile. Accounts that are configured by a profile, such as Exchange accounts, can be removed only by deleting the profile.

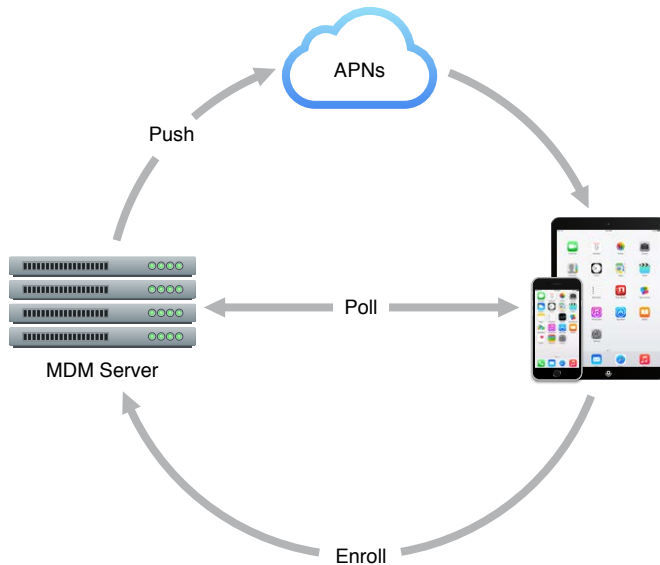
For more information, see [Configuration Profile Key Reference](#).

Mobile device management (MDM)

Overview

iOS and OS X support for MDM allows IT to securely configure and manage scaled Apple device deployments across their organizations. To accomplish this, iOS and OS X have a built-in MDM framework that lets third-party MDM solutions wirelessly interact with Apple devices. This lightweight framework was designed for Apple devices, and is powerful and scalable enough to configure and manage all the iOS, OS X, and Apple TV devices within an organization.

With an MDM solution in place, you can securely enroll Apple devices in an organization, configure and update settings, monitor compliance with organizational policies, and remotely wipe or lock managed devices. MDM for iOS and OS X gives you a simple way to let users access network services while ensuring Apple devices are properly configured—no matter who owns them.



MDM solutions use the Apple Push Notification Service (APNs) to maintain persistent communication with Apple devices across both public and private networks. MDM requires multiple certificates to operate, including an APNs certificate to talk to clients and an SSL certificate to communicate securely. MDM solutions may also sign profiles with a certificate. MDM capabilities are built on existing iOS and OS X technologies such as configuration profiles, over-the-air enrollment, and APNs. For example, APNs is used to wake the device so it can communicate directly with its MDM server over a secured connection.

Important: No confidential or proprietary information is transmitted via APNs.

MDM lets your IT department securely enroll personally owned and organization-owned Apple devices. With an MDM solution in place, you can configure and update settings, monitor compliance with organizational policies, and remotely wipe or lock managed Apple devices. MDM also enables distribution, management, and configuration of apps and books purchased through the Volume Purchase Program or developed in-house.

For more information about MDM, see:

- [Managing devices in education](#)
- [iOS and the new IT for enterprise](#)

Most certificates, including an APNs certificate, must be renewed annually. When a certificate expires, an MDM server can't communicate with Apple devices until the certificate is updated. Be prepared to update all MDM certificates before they expire. Contact your Certificate Authority (CA) for information about renewing your certificates. For more information about APNs, see [Apple Push Certificates Portal](#).

To enable management, Apple devices are enrolled with an MDM server using an enrollment configuration profile and can be done by the user directly. For company-owned devices, MDM enrollment can be automated using the Device Enrollment Program (described in this chapter). When an administrator initiates an MDM policy, option, or command, the Apple devices receive notification of the action through the APNs. With a network connection, devices can receive APNs commands anywhere in the world.

Enrollment

Enrolling Apple devices enables cataloging and asset management. The enrollment process typically leverages Simple Certificate Enrollment Protocol (SCEP), which lets a device create and enroll unique identity certificates for authentication to an organization's services.

In most cases, users decide whether or not to enroll their Apple device in MDM, and they can disassociate from MDM at any time. Organizations should consider incentives for users to remain managed. For example, require MDM enrollment for Wi-Fi network access by using the MDM solution to automatically provide the wireless credentials. When a user leaves MDM, their device attempts to notify the MDM server.

The Device Enrollment Program can also be used to automatically enroll Apple devices your organization owns in MDM during initial setup. You can also supervise the iOS devices, so users with these devices won't be able to bypass MDM or unenroll their devices.

For more information, see [Device Enrollment Program](#).

Configure

Once an Apple device is enrolled, it can be dynamically configured with settings and policies by the MDM server, which sends configuration profiles to the device that are automatically, and silently, installed by either iOS or OS X.

Configuration profiles can be signed, encrypted, and locked—preventing the settings from being altered or shared—ensuring that only trusted users and Apple devices that are configured to your specifications can access your network and services. If a user disassociates their device from MDM, all of the settings installed by MDM are removed.

A redesigned user interface for profiles in iOS 8 shows users what has been configured and restricted by MDM. Accounts, apps, books, and restrictions can now be easily viewed. Provisioning profiles are no longer visible to the user in iOS 8 and expired profiles are automatically removed.

Accounts

MDM can help your users get up and running quickly by setting up their mail and other accounts automatically. Depending on the MDM solution you use and its integration with your internal systems, account payloads can also be pre-populated with a user's name, mail address, and, where applicable, certificate identities for authentication and signing.

MDM can configure the following types of accounts:

- Calendar
- Contacts
- Exchange ActiveSync
- Identity
- Jabber
- LDAP

- Mail
- Subscribed Calendars
- VPN
- 802.1X

Managed mail and calendar accounts respect the Managed Open In restrictions in iOS 7 or later.

Queries

An MDM server has the ability to query Apple devices for a variety of information. This includes hardware information, such as serial number, device UDID, Wi-Fi MAC address, or FileVault encryption status (for OS X). It also includes software information, such as the device version, restrictions, and a detailed list of all apps installed on the device. This information can be used to help ensure that users maintain the appropriate set of apps. iOS and OS X allow queries about the last time a device was backed up to iCloud, and the app assignment account hash of the logged-in user.

With Apple TV software 5.4 or later, MDM can query enrolled Apple TV devices for asset information such as language, locale, and organization.

Management tasks

When an iOS device is managed, it can be administered by the MDM server through a set of specific tasks. Management tasks include:

- *Changing configuration settings:* A command can be sent to install a new or updated configuration profile on an Apple device. Configuration changes happen silently without user interaction.
- *Locking an iOS device:* If an iOS device needs to be locked immediately, a command can be sent to lock it with the current passcode.
- *Remotely wiping an iOS device:* If an iOS device is lost or stolen, a command can be sent to erase all of the data on it. Once a remote-wipe command is received, it can't be undone.
- *Clearing a passcode lock:* Clearing a passcode sets the iOS device so that it immediately requires the user to enter a new passcode. This is used when a user has forgotten their passcode and wants IT to reset it for them.
- *Clearing restrictions password:* Support for clearing the restrictions and restrictions password set on the iOS device by the user. This feature is available for supervised devices only.
- *Request AirPlay Mirroring:* Adds a command to prompt a supervised iOS device to begin AirPlay mirroring to a specific destination.
- *Stop AirPlay Mirroring:* Adds a command to prompt a supervised iOS device to stop AirPlay mirroring to a specific destination.

Certain tasks can be queued in iOS 8 or later and in OS X Mavericks or later, if the device is in Setup Assistant. Those tasks are:

- Invitation into the Volume Purchase Program (VPP)
- Install apps
- Install media
- Lock a device
- Request AirPlay mirroring (iOS only)

Managed apps

Distributing apps to your users can help them be more productive at work or in the classroom. However, depending on your organization's requirements, you may need to control how those apps connect to internal resources, and how data security is handled when a user transitions out of the organization—all while coexisting alongside the user's personal apps and data. Managed apps in iOS 7 or later and OS X Yosemite or later let your organization distribute free, paid, and in-house enterprise apps wirelessly via MDM, providing the right balance between institutional security and user personalization.

MDM servers can deploy apps from the App Store and apps developed in-house to Apple devices over the air. Both paid and free App Store apps can be managed by an MDM server using Volume Purchase Program (VPP) managed distribution. For more information about managed distribution with MDM, see the Volume Purchase Program [Overview](#).

Installing VPP apps can occur in the following ways:

- Users with a personal Apple device are prompted by MDM to install the app from the App Store using their Apple ID.
- On organizationally-owned supervised iOS devices enrolled with MDM, app installation occurs silently.

Managed apps can be removed remotely by the MDM server, or when the user removes their own Apple device from MDM. Removing the app also removes the data associated with the removed app. If the VPP app is still assigned to the user, or if the user redeemed an app code using a personal Apple ID, the app can be downloaded again from the App Store but won't be managed. If an app is revoked, it will continue to function for a limited time. Eventually the app is disabled and the user is informed that they need to purchase their own copy to continue using it.

iOS 7 added a suite of restrictions and capabilities to managed apps, providing improved security and a better user experience:

- *Managed Open In:* Provides two useful functions for protecting your organization's app data:
 - *Allow documents from unmanaged sources in managed destinations.* Enforcing this restriction prevents a user's personal sources and accounts from opening documents in the organization's managed destinations. For example, this restriction could prevent a user's copy of Keynote from opening a presentation PDF in an organization's PDF viewing app. This restriction could also prevent a user's personal iCloud account from opening an attachment in an organization's copy of Pages.
 - *Allow documents from managed sources in unmanaged destinations.* Enforcing this restriction prevents an organization's managed sources and accounts from opening documents in a user's personal destinations. This restriction could prevent a confidential email attachment in the organization's managed mail account from being opened in any of the user's personal apps.
- *App Configuration:* App developers can identify app settings that can be set when installed as a managed app. These configuration settings can be installed before or after the managed app is installed.
- *App Feedback:* App developers building apps can identify app settings that can be read from a managed app using MDM. For example, a developer could specify a "DidFinishSetup" key that an MDM server could query to determine if the app had been launched and set up.

- *Prevent Backup*: This restriction prevents managed apps from backing up data to iCloud or iTunes. Disallowing backup prevents managed app data from being recovered if the app is removed via MDM but later reinstalled by the user.

iOS 8 adds these management capabilities:

- *Safari downloads from managed domains*: Downloads from Safari are considered managed documents if they originate from a managed domain. For example, if a user downloads a PDF using Safari from a managed domain, that PDF complies with all managed document settings.
- *iCloud document management*: This restriction prevents managed apps from storing data in iCloud. For example, data created or used by a managed app can't be stored in iCloud, however data created by users in unmanaged apps can be stored in iCloud.

Restricting third-party keyboards

iOS 8 supports Managed Open In rules that apply to third-party keyboard extensions. This prevents unmanaged keyboards from appearing over managed apps.

Managed books

With iOS 8 and OS X Mavericks or later, you can distribute and manage books, eBooks, and PDFs that you create or purchase using MDM—allowing for seamless management of training materials and other business documents.

Books, eBooks, and PDFs distributed by MDM have the same properties as other managed documents—they can be shared only with other managed apps or mailed using managed accounts. Books purchased through the Volume Purchase Program can be distributed through managed book distribution, but can't be revoked and reassigned. A book already purchased by the user can't be managed, unless that book is explicitly assigned to the user using the Volume Purchase Program.

Managed domains

In iOS 8, you can manage specific URLs and subdomains. Any documents coming from those domains are also considered managed and will follow the behavior of the existing Managed Open In restrictions. Paths following the domain are managed by default. Alternate subdomains aren't included unless a wildcard is applied. Domains entered in Safari with www (for example, www.example.com) are treated as .example.com.

Shown in settings	Managed domains	Unmanaged domains
example.com	example.com/* www.example.com/*	*.example.com hr.example.com
example.com/docs	example.com/docs/* www.example.com/docs/*	example.com www.example.com hr.example.com/docs
www.example.com	www.example.com/* www.example.com/docs	example.com hr.example.com
*.example.com	*.example.com/*	example.com
*.example.com/docs	*.example.com/docs/*	example.com www.example.com

Profile Manager

In addition to third-party MDM solutions, Apple offers an MDM solution called Profile Manager, a service of OS X Server. Profile Manager makes it easy to configure Apple devices so they're set up to your organization's specifications.

Profile Manager provides three components:

- *Over-the-air configuration of Apple devices:* Streamline the configuration of institutionally-owned Apple devices. Enroll devices in MDM during activation and skip basic setup steps to get users up and running quickly.
- *Mobile device management service:* Profile Manager provides an MDM service that lets you remotely manage enrolled Apple devices. After a device is enrolled, you can update its configuration over the network without user interaction and perform other tasks.
- *App and book distribution:* Profile Manager can distribute apps and books purchased through the Volume Purchase Program (VPP). App and book assignment is supported on iOS devices with iOS 7 or later installed and Mac computers with OS X Mavericks or later installed.

For more information, see [Managing Devices](#). Also see [Profile Manager Help](#).

Supervise devices

To enable additional configuration options and restrictions, you should supervise iOS devices owned by your organization. For example, supervision lets you keep account settings from being modified, or lets you filter web connections via Global Proxy to make sure users' web traffic stays within the organization's network.

By default, all iOS devices are non-supervised. You can combine supervision and remote management with MDM to manage additional settings and restrictions. To enable supervision of your organization's devices, use Apple's Device Enrollment Program or Apple Configurator.

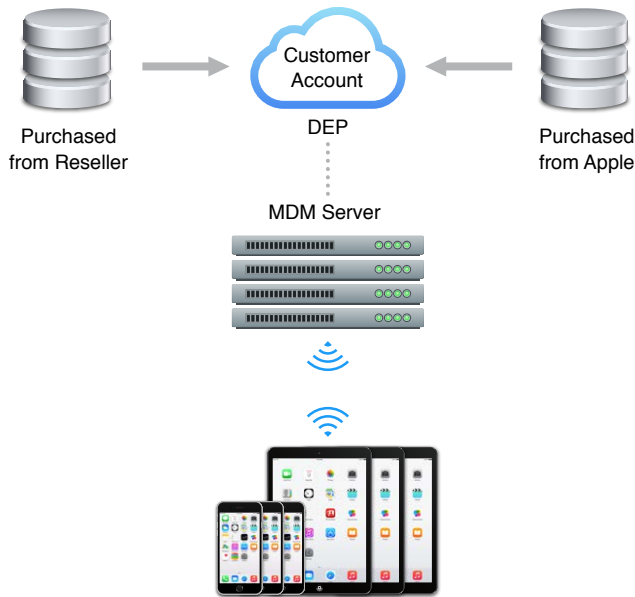
Supervision provides a higher level of device management for devices that are owned by your organization, allowing restrictions such as turning off iMessage or Game Center. It also provides additional device configurations and features, such as web content filtering, and the ability to silently install apps. With the Device Enrollment Program, supervision can be wirelessly enabled on the device as part of the setup process, or enabled manually using Apple Configurator.

For more information, see [Supervised settings](#).

Device Enrollment Program

The Device Enrollment Program (DEP) provides a fast, streamlined way to deploy Apple devices that your organization has purchased directly from Apple or participating Apple Authorized Resellers or carriers. You can automatically enroll the Apple devices in MDM without having to physically touch or prepare them before users get them. And you can further simplify the setup process for users by removing specific steps in Setup Assistant, so users are up and running quickly. You can also control whether or not the user can remove the MDM profile from the device. For example, you can order the devices from Apple or a participating Apple Authorized Reseller or carrier, configure all the management settings, and have the Apple devices shipped directly to the user's home address. Once the device is unboxed and activated, it enrolls in your MDM and all management settings, apps, and books are ready for the user.

The process is simple: After enrolling in the program, administrators log into the DEP website, link the program to their MDM server or servers, and “claim” the Apple devices purchased from Apple or a participating Apple Authorized Reseller or carrier. The devices can then be assigned to an MDM server. Once the device is enrolled, any MDM-specified configurations, restrictions, or controls are automatically installed.



Apple devices must meet the following criteria in order to be eligible for assignment using the Device Enrollment Program:

- Devices must be ordered on or after March 1, 2011 and purchased directly from Apple using your enrolled and verified Apple customer numbers.
- Devices must be purchased directly from a participating Apple Authorized Reseller or carrier and linked to that reseller’s DEP Reseller ID. The actual date of eligibility is determined by your participating Apple Authorized Reseller or carrier’s sales history, but the date can’t be before March 1, 2011.

Note: The Device Enrollment Program isn’t available in all countries or regions.

Eligible Apple devices are available for assignment by order number to your MDM servers on the [Apple Deployment Programs](#) website. You can also look up devices by type and by serial number within those orders. After new orders ship, you can search for them on the DEP website and automatically assign them to a specific MDM server. For example, when you place an iPad order for 5000 units, you can use the order number to assign all or a specific number of the iPad devices to an existing authorized MDM server. You can also assign devices to a specific MDM server by serial number. This method is helpful in situations where the devices you need to assign are in your physical possession.

Future orders can be automatically assigned to an authorized MDM server, negating the need for you to manually manage Apple device assignment.

After a device has been assigned to an MDM server in the program, profiles and additional features may be applied using your organization's MDM server. These features include:

- Enable supervision
- Mandatory configuration
- Requiring authentication to your directory system to complete setup
- Lockable MDM enrollment profile
- Skipping steps in Setup Assistant

For more information, see:

- [Apple Deployment Programs](#)
- [Device Enrollment Programs Help](#)
- [Device Enrollment Program for Education](#)

Apple Configurator

For iOS devices that are managed by you and not set up by individual users, you can use Apple Configurator, a free Mac app available on the App Store. It lets you set up and configure multiple devices at a time with USB, before you give them to users. With this tool, your organization can quickly configure and update multiple devices to the latest version of iOS, configure device settings and restrictions, and install apps and content. You can also restore a backup to devices, which applies device settings and the Home screen layout, and installs app data.

Apple Configurator is ideal whenever users share iOS devices that need to be quickly refreshed and kept up to date with the correct settings, policies, apps, and data. You can also use Apple Configurator to supervise devices before using MDM to manage settings, policies, and apps.

For more information, see [Apple Configurator Help](#).

App and book distribution

8

Overview

iOS comes with a collection of powerful built-in apps that let people in your organization easily accomplish everyday tasks—from managing email and calendars to keeping track of contacts and web content. And the additional functionality users need in order to be productive comes from the hundreds of thousands of third-party apps available on the App Store, or from custom enterprise apps developed in-house or by third-party developers. In education, you can keep your users productive and creative with relevant iOS apps and content.

There are several ways to deploy apps and books to Apple devices throughout your organization. The most scalable method is to purchase apps and books through the Volume Purchase Program (including B2B apps) and assign them to users with MDM. Your organization can also create and deploy its own in-house apps by joining the iOS Developer Enterprise Program. If you're in a shared-device deployment model, you can install apps and content locally with Apple Configurator.

When deploying apps and books, consider the following:

- Volume Purchase Program (VPP)
- Custom B2B apps
- Apps and books developed in-house
- Deploying apps and books
- Caching Server

Volume Purchase Program (VPP)

Overview

The App Store and iBooks Store feature thousands of great apps and books that users can purchase, download, and install. The Volume Purchase Program (VPP) gives organizations a simple way to purchase apps and books in volume and distribute them to employees, contractors, teachers, or students. All paid and free apps and books on the App Store and iBooks Store are eligible for purchase under the program. There are two Volume Purchase Program websites, one for business, and one for education.

VPP for Business lets you get custom B2B apps, custom-built for you by third-party developers and downloaded privately through the VPP store.

VPP for Education lets app developers offer special pricing for purchases of 20 apps or more to eligible educational institutions, including any K–12 institution or district, or any accredited, degree-granting higher education institution. Special pricing isn't available for books.

MDM solutions can be integrated with VPP, enabling your organization to purchase apps and books in volume and assign them to specific users or groups. When a user no longer needs an app, you can use MDM to revoke and reassign it to a different user. And each app or books is automatically available for download on user's Apple device. Once distributed, books remain the property of the recipient and aren't revocable or reassignable.

For more information, see:

- [Volume Purchase Program for Business](#)
- [Volume Purchase Program Education](#)
- [Apple Deployment Programs Help](#)

Note: The Volume Purchase Program isn't available in all countries or regions.

Enroll in the Volume Purchase Program

To purchase apps in volume, you first need to enroll and create an account with Apple. You need to provide information about your organization, such as a D&B D-U-N-S number (if you're a business) and contact information. You also need to create an Apple ID that's used only to administer this program.

Purchase apps and books in volume

You use the [Volume Purchase Program](#) website to purchase apps and books for your business or educational institution.

Use the Apple ID associated with your Volume Purchase Program account to log in to the website. Search for the apps or books you want to purchase, then indicate the number of copies you're purchasing. You can pay with a corporate credit card or VPP Credit that you've procured using a purchase order (PO). There's no limit to the number of copies of an app or book that you can purchase. Apps and books are then available for assignment via your MDM solution, provided it's linked to your VPP account and has a valid token.

Managed distribution

When you buy apps and books in volume, you distribute them via MDM using managed distribution to assign apps and books to users on iOS 7 or later or OS X Mavericks or later. When they no longer need the app or when they leave your organization, you can reassign it to a different user. Books cannot be revoked after they're assigned.

Before MDM is used to assign apps to users, you'll need to link your MDM server to your VPP account using a secure token. You can download this secure token to your MDM server by accessing your account summary from the VPP store. For more information, see [Apple Deployment Programs Help](#).

In order for users to participate in managed distribution via VPP, they must first be invited. When a user accepts an invitation to participate in managed distribution, their personal Apple ID is linked to your organization. The user doesn't need to tell you what their Apple ID is, and there's no need for you to create and provide Apple IDs for their use. For education accounts, you can create Apple IDs for students under the age of 13. For more information, see the [Apple ID for Students](#) website.

It's important to register and assign apps and books to VPP users in advance of sending an invitation to the user. This provides more time for the assignment to propagate to the user's purchase history upon accepting the VPP invitation. Registering users and assigning apps and books can happen at any time, including before the Apple devices are enrolled in MDM.

Once an app is assigned to a user via MDM, it appears in the purchase history of the App Store for that user. The user can be prompted to accept installation of the app or, in the case of a supervised iOS device, the app can be silently installed.

If any apps that aren't already installed on a device are pushed using the Push VPP Apps task, they'll be automatically removed when a user unenrolls from MDM.

Apps and books distributed with managed distribution are not shared with family members when using Family Sharing.

Custom B2B apps

Custom apps that a developer creates or customizes for your business (B2B) can also be purchased via the Volume Purchase Program.

Developers registered in the iOS Developer Program can submit apps for B2B distribution using iTunes Connect, which is the same process used to submit other apps to the App Store. The developer sets the price per copy and adds your Volume Purchase Program Apple ID to their authorized B2B purchasers list. Only authorized purchasers are able to see or purchase the app.

B2B apps aren't secured by Apple—the security of the data in an app is the responsibility of the developer. Apple recommends using iOS best practices for in-app authentication and encryption.

After Apple reviews the app, you use the Volume Purchase Program website to purchase copies as described in [Purchase apps and books in volume](#). Custom B2B apps aren't listed on the App Store—you purchase them through the [Volume Purchase Program](#) website.

In-house apps

Develop iOS apps for use by your organization's employees using the iOS Developer Enterprise Program. This program offers a complete and integrated process for developing, testing, and distributing your iOS apps to employees within your organization. You can distribute in-house apps either by hosting the apps on a simple, secure web-server you create internally, or by using a third-party MDM or app management solution.

The benefits of managed apps with MDM include the ability to configure apps remotely, manage versions, configure Single Sign-On, set policies for network access, and control which apps can export documents. Your specific requirements, infrastructure, and level of app management will dictate which solution makes the most sense for you.

To develop and deploy in-house apps for iOS:

- 1 Register for the iOS Developer Enterprise Program.
- 2 Prepare your app for distribution.
- 3 Create an enterprise distribution provisioning profile that authorizes devices to use apps you've signed.
- 4 Build the app with the provisioning profile.
- 5 Deploy the app to your users.

Register for app development

Once you register for the iOS Developer Enterprise Program, you can request a developer certificate and developer provisioning profile. You use these during development to build and test your app. The development provisioning profile lets apps signed with your developer certificate run on registered iOS devices. The ad hoc profile expires after three months and specifies which devices (by device ID) can run development builds of your app. Distribute your developer-signed build and the development provisioning profile to your development team and app testers.

For more information, see the [iOS Developer Enterprise Program](#) website.

Prepare apps for distribution

After you finish development and testing and are ready to deploy your app, you sign your app using your distribution certificate and package it with a provisioning profile. The designated Team Agent or the Admin for your program membership creates the certificate and profile at the [iOS Dev Center](#) website.

In iOS 8, provisioning profiles are no longer viewable by the user on their iOS device.

Provision in-house apps

The enterprise distribution provisioning profile lets your app be installed on an unlimited number of iOS devices. You can create an enterprise distribution provisioning profile for a specific app, or for multiple apps.

Once you have both the enterprise distribution certificate and provisioning profile installed on your Mac, you use Xcode to sign and build a production version of your app. Your enterprise distribution certificate is valid for three years, and you can have up to two valid certificates at one time. When a certificate is ready to expire, you have to sign and build your app again using a renewed certificate. The provisioning profile for the app is good for one year, so you should release new provisioning profiles annually. For more information, see [Provide updated apps](#).

It's important that you limit access to your distribution certificate and its private key. Use Keychain Access on OS X to export and back up these items in .p12 format. If the private key is lost, it can't be recovered or downloaded a second time. You should also restrict access to personnel who are responsible for final acceptance of the app. Signing an app with the distribution certificate gives your organization's seal of approval to the app's content, function, and adherence to the Enterprise Developer Agreement licensing terms.

For more information about deploying in-house apps, see Apple's [App distribution guide](#).

In-house books

iOS 8 and OS X Yosemite bring major enhancements with the introduction of managed distribution for books. This feature lets you assign books to users using MDM so the books remain under the control of your organization. PDFs and ePubs you create can be assigned to users, revoked, and reassigned to different users when no longer needed, just like in-house apps.

Deploy apps and books

Overview

You can use the following ways to deploy apps and books:

- Use your MDM server to instruct managed Apple devices to install an in-house or App Store app, if your MDM server supports it.
- Post the app on a secure web server, so users can access and perform the installation wirelessly. For information, see [Install in-house apps wirelessly](#).
- You can install the app on iOS devices locally using Apple Configurator.

Install apps and books using MDM

An MDM server can manage third-party apps from the App Store, as well as in-house apps. Apps installed using MDM are called *managed apps*. The MDM server can specify whether managed apps and their data remain when the user unenrolls from MDM. The server can prevent managed app data from being backed up to iTunes and iCloud. This lets you manage apps that may contain sensitive business information with more control than apps downloaded directly by the user.

In order to install a managed app, the MDM server sends an installation command to the Apple device. If the app is from the App Store, it will be downloaded and installed from Apple. If it's an in-house it will be installed from your MDM solution. On unsupervised devices, managed apps require a user's acceptance before they're installed.

On iOS 7 or later and OS X Mavericks or later, VPN connections can be specified at the app layer, so only the network traffic for that app is in the protected VPN tunnel. This ensures that private data remains private, and public data doesn't get mixed with it.

Managed apps support Managed Open In in iOS 7 or later. This means that managed apps can be restricted from transferring data to or from the user's personal apps, which lets your organization ensure that sensitive data remains where it needs to be.

An MDM server can install books from the iBooks Store that you have assigned to the user using VPP. It can also install managed PDFs, eBooks, and iBooks Author books from your own servers, and update them with newer versions as needed. The server can prevent managed books from being backed up. Managed books will be removed when the user unenrolls from MDM.

Install apps with Apple Configurator

Apple Configurator simplifies basic setup and configuration tasks—but it can also be used to install apps and other content on iOS devices. Apple Configurator is most helpful when it's used to supervise devices that won't be personalized by the user, such as shared iPad devices in a classroom.

In addition to apps, you can use Apple Configurator to install documents so they're available when your users start using the devices. Documents are available for apps that support iTunes file sharing. You can also review or retrieve documents from iOS devices by connecting them to a Mac running Apple Configurator.

Caching Server

iOS and OS X make it easy for users to access and consume digital content, and some users may request many gigabytes of apps, books and software updates while connected to an organization's network. The demand for these assets comes in spikes—first with initial Apple device deployment, and then sporadically, as users discover new content or as content is updated over time. These content downloads can cause surges in demand for Internet bandwidth.

Caching Server is a service of OS X Server that saves previously requested content on your organization's local network. This reduces the bandwidth needed to download content. It does this by reducing outbound Internet bandwidth on private networks ([RFC 1918](#)) and storing cached copies of requested content on the local area network.

Caching Server on OS X Server Yosemite caches the following types of content for iOS 7 or later and OS X Mountain Lion v10.8.2 or later:

- iOS software updates (iOS 8.1 or later)
- OS X software updates
- Internet Recovery image (OS X Mavericks or later)
- Java and print driver updates
- App Store apps
- App Store updates
- Books from the iBooks Store
- iTunes U courses and content
- GarageBand downloadable content
- High-quality voices and language dictionaries

For more information about content cached by the caching service, see the Apple Support article [OS X Server: Content types supported by the Caching service](#).

iTunes also supports Caching Server. The following types of content are supported by iTunes 11.0.4 or later (on both Mac and Windows computers):

- App Store apps
- App Store updates
- Books from the iBooks Store

Large networks benefit from having multiple Caching Servers in place. For many deployments, configuring Caching Server is as simple as turning on the service. With Caching Server on OS X Server Yosemite, a NAT environment for the server and all devices that use it is no longer required. Caching Server can now be used on networks consisting of publicly routable IP addresses. Apple devices running iOS 7 or later and OS X Mountain Lion v10.8.2 or later automatically contact a nearby Caching Server without any additional configuration.

For more information, see [Caching Service](#) in OS X Server Help.

Here's an explanation of the Caching Server workflow:

- 1 When an Apple device on a network with one or more Caching Servers requests content from the iTunes Store or Software Update server, the device is referred to a Caching Server.
- 2 The Caching Server first checks to see whether it already has the requested content in its local cache.
 - If it does, it immediately begins serving the content to the device.
 - If the Caching Server doesn't have the requested asset, it attempts to download the content from another source. Caching Server 2 or later includes a peer replication feature that can use other Caching Servers on the network, if those servers have already downloaded the requested content.
- 3 As the Caching Server receives download data, it immediately relays the data to any devices that have requested it and simultaneously caches a copy on the designated storage device.

Planning for support

9

Overview

A deployment of Apple devices should include support. AppleCare offers support plans for organizations of all sizes, including software deployment support and hardware coverage:

- Mac computers with and OS X, iOS devices with iOS
 - AppleCare Help Desk Support
 - AppleCare OS Support
 - AppleCare for Enterprise
- Just iOS devices
 - AppleCare for iOS device users
 - iOS Direct Service Program
- Just Mac computers
 - AppleCare Protection Plan for Mac or Apple Display

You can choose the plans that fit your organization's needs. For more information, see the [AppleCare Professional Support](#) website.

AppleCare Help Desk Support

AppleCare Help Desk Support provides priority telephone access to Apple's senior technical support staff. It includes a suite of tools to diagnose and troubleshoot Apple hardware, which can help institutions manage their resources more efficiently, improve response time, and reduce training costs. AppleCare Help Desk Support covers an unlimited number of support incidents for hardware and software diagnosis and troubleshooting, and issue isolation for iOS and OS X devices. Learn more at the [AppleCare Help Desk Support](#) website.

AppleCare OS Support

AppleCare OS Support includes [AppleCare Help Desk Support](#), in addition to incident support. AppleCare OS Support includes support for system components, network configuration, and administration; integration into heterogeneous environments; professional software applications, web applications, and services; and technical issues requiring the use of command-line tools for resolution. Learn more at the [AppleCare OS Support](#) website.

AppleCare for Enterprise

AppleCare for Enterprise includes comprehensive hardware and software for your business or educational institution. Your AppleCare Account Manager will help review your IT infrastructure, track issues you may be having, and provide monthly activity reports for both support calls and repairs. You'll get IT department-level support by phone or email for all Apple hardware and software. We'll provide support for complex deployment and integration scenarios, including MDM and Active Directory. And if you need help with IBM MobileFirst for iOS apps, we'll help troubleshoot your solution and work with IBM to get your issue resolved. AppleCare for Enterprise can help reduce the load on your internal help desk by providing technical support for your employees over the phone, 24/7. Apple provides technical support for Apple hardware and operating systems, for Apple apps such as Keynote, Pages, and Numbers, and for personal accounts and settings. For more information, see the [AppleCare for Enterprise](#) website.

AppleCare for iOS device users

Every iOS device comes with a one-year limited warranty and complimentary telephone technical support for 90 days after the purchase date. This service coverage can be extended to two years from the original purchase date with [AppleCare+ for iPhone](#), [AppleCare+ for iPad](#), or [AppleCare+ for iPod touch](#). Users can call Apple's technical support experts as often as they like for basic usage support. Apple also provides convenient service options when devices need to be repaired. All three programs offer up to two incidents of accidental damage coverage, each subject to a service fee.

iOS Direct Service Program

As a benefit of AppleCare+ and the AppleCare Protection Plan, the iOS Direct Service Program lets your help desk screen devices for issues without calling AppleCare or visiting an Apple Store. If necessary, your organization can directly order a replacement iPhone, iPad, iPod touch, or in-box accessories. Learn more at the [iOS Direct Service Program](#) website.

AppleCare Protection Plan for Mac or Apple Display

Every Mac comes with a one-year limited warranty and complimentary telephone technical support for 90 days after the purchase date. This service coverage can be extended up to three years from the original purchase date and may include onsite repair for desktop computers. Mail-in repair for portable computers and Apple displays and carry-in repair for Mac computers or Apple displays at an Apple Retail Store or other Apple Authorized Service Provider is also included. You can call Apple's technical support experts as often as you like with questions about Apple hardware, OS X, and Apple apps such as those contained in the iLife and iWork suites.

Restrictions

Overview

Apple devices support the following policies and restrictions, which you can configure to meet the needs of your organization. Depending on your MDM solution, the names of these restrictions may vary slightly.

Note: Not all restrictions are available for all Apple devices.

Device Enrollment Program settings

The following restrictions apply to Apple devices assigned to MDM servers using the Device Enrollment Program.

Enrollment options

- *Prompt user to enroll device:* When this option is on, the device prompts the user to enroll the device in MDM. The default is off.

The following settings are subsets of the one above.

- *Do not allow user to skip enrollment step:* When this option is on, the user must enroll the device in MDM before they can set up the device. The default is off.
- *Require credentials for enrollment:* When this option is off, users do not need to authenticate to a directory service before they enroll the device in MDM. The default is on.
- *Supervise device (iOS-only):* When this option is on, the device is supervised during enrollment and can't be unenrolled by the user. The default is off.

The following setting is a subset of the one above.

- *Allow pairing (iOS-only):* When this option is off, users can't pair their device with a computer. The default is on.
- *Prevent unenrollment:* When this option is on, a supervised iOS device can't be unenrolled by the user. Mac computers can be unenrolled if the user has the administrator user name and password. The default is off.

Setup Assistant options

Apple devices enrolled in the Device Enrollment Program and managed by MDM can have these Setup Assistant screens skipped (the default for all options is on):

Unless these items are also permanently restricted using the MDM solution, users can perform any of these after the Apple device is set up.

- *Set up as a new device or restore from backup:* When this option is off, the user can't select between the two choices.
- *Allow user to enter their Apple ID:* When this option is off, the user can't enter their Apple ID.
- *Allow user to view the Terms and Conditions:* When this option is off, the user can't review the Apple Terms and Conditions.

- *Allow user to select whether diagnostic data is sent to Apple and developers:* When this option is off, the user can't select whether to send diagnostic data to Apple and app data to developers.
- *Allow user to enable Location Services:* When this option is off, the user can't enable Location Services.
- *Allow the user to enable Touch ID (iOS-only):* When this option is off, the user can't enable Touch ID to unlock the device or authenticate to apps that use Touch ID.
- *Allow user to alter the Passcode Lock settings (iOS-only):* When this option is off, the user can't alter the passcode from the managed setting.
- *Allow user to enable Apple Pay (iOS-only):* When this option is off, the user can't enable Apple Pay.
- *Allow user to enable Siri (iOS-only):* When this option is off, the user can't enable Siri.
- *Allow user to change the Display Zoom (iOS-only):* When this option is off, the user can't select from the standard or zoomed Display Zoom setting.
- *Allow the user to register the Mac with Apple (OS X-only):* When this option is off, the user can't fill out the registration form and send it to Apple.
- *Allow the user to enable FileVault (OS X-only):* When this option is off, the user can't enable FileVault.

Device functionality

iOS device functionality

The following are iOS functionality restrictions:

- *Allow installing apps:* When this option is off, App Store is disabled and its icon is removed from the Home screen. Users can't install or update apps from the App Store using App Store, iTunes, or MDM. In-house apps can still be installed and updated.
- *Allow Siri:* When this option is off, Siri can't be used.
- *Allow Siri while locked:* When this option is off, Siri only responds when the device is unlocked.
- *Apple Pay:* When this option is off, Apple Pay is disabled.
- *Allow Handoff:* When this option is off, users can't use Handoff with their Apple devices.
- *Allow use of camera:* When this option is off, cameras are disabled and the Camera icon is removed from the Home screen. Users can't take photographs or videos, or use FaceTime.
- *Allow FaceTime:* When this option is off, users can't place or receive FaceTime audio or video calls.
- *Allow screenshots:* When this option is off, users can't save a screenshot of the display.
- *Allow automatic syncing while roaming:* When this option is off, devices that are roaming sync only when an account is accessed by the user.
- *Allow voice dialing:* When this option is off, users can't use voice commands to dial their phone.
- *Allow In-App Purchase:* When this option is off, users can't make in-app purchases.
- *Allow Touch ID to unlock device:* When this option is off, users must use a passcode to unlock the device.
- *Force Apple Watch wrist detection:* When this option is on, Apple Watch locks automatically when it's removed from the user's wrist. It can be unlocked with its passcode or the paired iPhone. The default is Off.
- *Allow Control Center access from Lock Screen:* When this option is off, users can't swipe up to view Control Center.

- *Allow Notification Center access from Lock Screen:* When this option is off, users can't swipe down to see Notification Center in the Lock screen.
- *Allow Today view from Lock Screen:* When this option is off, users can't swipe down to see Today View in the Lock screen.
- *Allow Passbook notifications in Lock Screen:* When this option is off, users must unlock the device to use Passbook.
- *Require iTunes password for all purchases:* When this option is off, iTunes in-app purchases and iTunes purchases don't prompt for the account password.
- *Set allowed content ratings:* Sets the region and ratings for movies, TV shows, and apps.

Mac functionality

The following are Mac functionality restrictions:

- *Allow App Store adoption:* When this option is off, iLife and iWork apps that shipped with OS X can't be adopted by the App Store.
- *Restrict App Store to software updates only:* When this option is on, the App Store can only be used to update apps. The default is Off.
- *Require admin password to install or update apps:* When this option is on, an administrator password is required in order to update any apps. The default is Off.
- *Restrict which apps are allowed to launch:* When this option is on, you can restrict which apps can be used. The default is Off.
- *Restrict specific System Preferences panes:* When this option is on, you can select which items in System Preferences users can access. If a pane isn't listed, make sure it's installed on the Mac with Profile Manager installed. The default is Off.
- *Lock the Desktop picture:* When this option is on, you can prevent the user from changing the Desktop picture. The default is Off.

Supervised settings

The two Activation Lock restrictions are off by default for all users and user groups.

- *Send "Allow Activation Lock" command after MDM enrollment:* When this option is on users are allowed to configure their iOS device so it can't be erased without entering the user's Apple ID.

The following setting is a subset of the above.

- *Only send command if Activation Lock bypass code has been obtained:* When this option is on, the MDM server must receive an Activation Lock bypass code before the user can configure their iOS device so it can't be erased without entering the user's Apple ID.
- *Global network proxy for HTTP:* When this payload is added to a profile, iOS devices must use the proxy defined in the payload for all network traffic using HTTP.
- *Allow iMessage:* When this option is off for Wi-Fi-only devices, the Messages app is hidden. When this option is off for device with Wi-Fi and cellular, the Messages app is still available, but only the SMS/MMS service can be used.
- *Allow Game Center:* When this option is off, the Game Center app and its icon are removed.
- *Allow removal of apps:* When this option is off, users can't remove installed apps.
- *Allow iBooks Store:* When this option is off, users can't purchase books through the iBooks Store.
- *Allow Podcasts:* When this option is off, users can't download Podcasts.
- *Allow predictive keyboard:* When this option is off, users won't see the predictive keyboard.
- *Allow auto correction:* When this option is off, users won't see any word correction suggestions.

- *Allow spell check*: When this option is off, users won't see potentially misspelled words underlined in red text.
- *Allow Define*: When this option is off, users can't double-tap to search for a word's definition.
- *Allow user-generated content in Siri*: When this option is off, Siri can't obtain content from sources that allow user-generated content, such as Wikipedia.
- *Allow manual install of configuration files*: When this option is off, configuration profiles can't be manually installed by users.
- *Allow configuring restrictions*: When this option is off, users can't set their own restrictions on their device.
- *Allow pairing to computers for content sync*: When this option is off, users can't pair their iOS device with anything but the Mac with Apple Configurator installed, where the device was first supervised.
- *Allow AirDrop*: When this option is off, users can't use AirDrop with any apps.
- *Allow the modification of Touch ID fingerprints*: When this option is off, users can't add or remove existing Touch ID information.
- *Allow account modification*: When this option is off, users can't create new accounts or change their user name, password, or other settings associated with their account.
- *Allow cellular data settings modification*: When this option is off, users can't change any settings regarding what apps use cellular data.
- *Allow Find My Friends settings modification*: When this option is off, users can't change any settings in the Find My Friends app.
- *Allow Erase All Content and Settings*: When this option is off, users can't erase their device and reset it to factory defaults.
- *Allow specific URLs (Content Filter payload)*: When this payload is added to a profile, users can't choose which websites they can view on their iOS devices.
 - *Permitted URLs*: Add URLs to this list to allow access to certain websites, even if they're considered adult by the automatic filter. If you leave this list empty, access is allowed to all non-adult websites except for those listed in Blacklisted URLs.
 - *Blacklisted URLs*: Add URLs to this list to deny access to certain websites. Users can't visit these sites even if they're considered non-adult by the automatic filter.
 - *Specific web sites only*: The user of the device will only have access to the web sites you define. Enter the URL of the website in the URL column. Enter the name for the bookmark in the Name column. To create a bookmark in a folder, enter the location of the folder in the Bookmark column. For example, create a bookmark in the Favorites folder by entering / Favorites/.
- *Restrict AirPlay connections with whitelist and optional connection passcodes*: When this option is off, a passcode isn't required when a device is first paired for AirPlay.
- *Enable Siri Profanity Filter*: When this option is off, the profanity filter in Siri isn't enabled.
- *Single App Mode*: Allows only a single, selected app to be used.
- *Accessibility settings*: Allows for certain accessibility settings when in Single App Mode.

For more information about iOS device supervision, see [Supervise devices](#).

Security and privacy settings

iOS and OS X security and privacy settings

The following security and privacy restriction is for both iOS and OS X:

- *Allow diagnostic data to be sent to Apple:* When this option is off, diagnostic data about a device will not be sent to Apple.

iOS security and privacy settings

The following security and privacy restrictions are iOS-only:

- *Allow Internet search results in Spotlight:* When this option is off, Spotlight won't return any results from an Internet search.
- *Domains (email):* Mail messages that aren't addressed to domains in the approved list will be marked. For example, a user could have both example.com and group.example.com in their list of known domains. If the user entered anyone@foo.com, that address would be marked so the user would clearly know the domain wasn't in the list.
- *Domains (Safari):* Downloads from Safari will be considered managed documents if they originate from a managed domain. For example, if a user downloads a PDF using Safari from a managed domain, that PDF will comply with all managed document settings.
- *Allow documents from unmanaged sources in managed destinations:* When this option is off, documents created or downloaded from unmanaged sources can't be opened in managed destinations.
- *Allow documents from managed sources in unmanaged destinations:* When this option is off, documents created or downloaded from managed sources can't be opened in unmanaged destinations.
- *Allow explicit music, Podcasts, and iTunes U:* When this option is off, explicit music or video content purchased from the iTunes Store or listed in iTunes U is hidden. Explicit content is flagged by content providers, such as record labels, when sold through the iTunes Store or distributed through iTunes U.
- *Allow erotica from iBooks Store:* When this option is off, explicit sexual content purchased from the iBooks Store is hidden. Explicit content is flagged by content providers when sold through the iBooks Store. Requires supervision for iOS 6.

Management of allowing explicit music, Podcasts, iTunes U content, content ratings themselves, and erotica from the iBooks Store are in both iTunes and the iBooks apps in OS X.

- *Allow automatic updates to certificate trust settings:* When this option is off, automatic updates to certificate trust settings can't take place.
- *Allow untrusted TLS certificates:* When this option is off, users aren't asked if they want to trust certificates that can't be verified. This setting applies to Safari, Mail, Contacts, and Calendar accounts. When this option is on, only certificates with trusted root certificates are accepted without a prompt. For information about Root CAs accepted by iOS, see the Apple Support article [List of available trusted root certificates](#).
- *Require passcode on first AirPlay pairing:* When this option is off, a passcode isn't required when a device is first paired for AirPlay.
- *Force limited ad tracking:* When this option is off, apps can use the Advertising Identifier (a nonpermanent device identifier) to serve the user targeted ads.
- *Allow backup of enterprise books:* When this option is off, users can't back up books distributed by their organization to iCloud or iTunes.
- *Force encrypted backups:* When this option is off, users can choose whether or not device backups performed in iTunes are stored in encrypted format on their Mac.

If any profile is encrypted and this option is turned off, encryption of backups is required and enforced by iTunes. Profiles installed on the device by Profile Manager are never encrypted.

OS X security and privacy

The following security and privacy restriction is for OS X-only:

- *Allow AirDrop*: When this option is off, users can't use AirDrop with other Mac computers. You can restrict the use of AirDrop for iOS devices, however they must be supervised first.

App usage

iOS and OS X app restrictions

The following app restrictions are for both iOS and OS X:

- Mail app restrictions:
 - *Allow Mail messages to be moved from one account to another*: When this option is off, users can't move a mail message from one account to another.
 - *Use only in Mail*: When this option is off, other apps can be used to send in-app mail from the specified account.
 - *Allow syncing of recent Mail addresses*: When this option is off, recently used addresses aren't synced across devices.
- Safari app restrictions:
 - *Allow Safari autofill*: When this option is off, Safari doesn't remember what users enter in web forms.
- Game Center restrictions:
 - *Allow multiplayer gaming*: When this option is off, users can't play multiplayer games in Game Center.
 - *Allow adding Game Center friends*: When this option is off, users can't find or add friends in Game Center.

iOS-only app restrictions

The following app restrictions are for iOS:

- iTunes Store restrictions:
 - *Allow use of iTunes Store*: When this option is off, the iTunes Store is disabled and its icon is removed from the Home screen. Users can't preview, purchase, or download content.
- Safari app restrictions:
 - *Allow use of Safari*: When this option is off, the Safari web browser app is disabled and its icon is removed from the Home screen. This also prevents users from opening web clips.
 - *Receive forced fraud warnings*: When this option is off, Safari will not attempt to prevent the user from visiting websites identified as being fraudulent or compromised.
 - *Enable JavaScript*: When this option is off, Safari ignores all JavaScript on websites.
 - *Block pop-ups*: When this option is off, pop-ups will not be blocked in Safari.
 - *Edit cookie preferences*: Sets the cookie policy in Safari. Choose to always block all cookies, always accept all cookies, allow cookies from current websites only, or from websites the user visits. The default is Always.

OS X-only app restrictions

The following app restrictions are for OS X:

- Dashboard app restrictions:

- *Allow specific Dashboard widgets to run:* When this option is on, you can select which Dashboard widgets the user can enable.
- Game Center restrictions:
 - *Allow Game Center:* When this option is off, the Game Center app and its icon are removed.
 - *Allow Game Center account modification:* When this option is off, users of Game Center can't modify their user name or password.

iCloud settings

- *Allow backup:* When this option is off, device backup is only performed in iTunes.
- *Allow document and data sync:* When this option is off, documents and data aren't added to iCloud.
- *Allow keychain sync:* When this option is off, iCloud Keychain is not used.
- *Allow My Photo Stream:* When this option is off, photos in My Photo Stream are erased from the device, photos from the Camera Roll aren't sent to My Photo Stream, and photos and videos in shared streams can no longer be viewed on the device. If there are no other copies of these photos and videos, they may be lost.
- *Allow iCloud photo sharing:* When this option is off, users can't subscribe to or publish shared photo streams.
- *Allow managed apps to store data in iCloud:* When this option is off, users can't store data from managed apps in iCloud.
- *Allow notes and highlights sync for enterprise books:* When this option is off, users can't sync notes or highlights to other devices using iCloud.

Profile Manager user and user group restrictions

These settings are on by default for all users and the user group "Everyone." They are off by default for the group, "Workgroup" and any administrator-created user groups.

Other MDM solutions may have similar settings, however the description of the setting will vary.

- *Allow access to My Devices portal:* When this option is on, user can access the Profile Manager My Devices portal.

The following are subsets of this setting.

- *Allow configuration profile downloads:* When this option is on, users can download configuration profiles from the My Device portal.
- *Allow device enrollment and unenrollment:* When this option is on, users can enroll additional devices and unenroll devices.
- *Allow device wipe:* When this option is on, users have the ability to wipe their devices.
- *Allow device lock (iOS-only):* When this option is on, users can lock their iOS device.
- *Allow device passcode to be cleared (iOS-only):* When this option is on, users can clear their iOS device passcode.
- *Allow enrollment during Setup Assistant for devices configured using the Device Enrollment Program:* When this option is on, Apple devices in the Device Enrollment Program assigned to this instance of Profile Manager can enroll their device in Profile Manager's MDM service.

The following are off by default for all users and user groups.

- *Allow enrollment during Setup Assistant for devices configured using the Apple Configurator (iOS-only):* When this option is on, iOS devices set up with Apple Configurator can enroll their device in Profile Manager's MDM service.

- *Restrict enrollment to placeholder devices:* When this option is on, only devices that have a placeholder with one of the following can enroll in Profile Manager's MDM service:
 - Serial Number
 - UDID
 - IMEI
 - MEID
 - Bonjour device ID (Apple TV only)

Note: The following is a subset of this restriction.
- *Restrict enrollment to assigned devices:* When this option is on, only devices that have been assigned to a user can enroll in Profile Manager's MDM service.

Install in-house apps wirelessly

Both iOS and OS X support over-the-air installation of custom in-house apps without using iTunes or the App Store.

Requirements:

- An iOS app in .ipa format, built for production with an enterprise provisioning profile
- An XML manifest file, described in this appendix
- A network configuration that lets devices access an iTunes server at Apple
- The use of HTTPS for iOS 7.1 or later

Installing the app is simple. Users download the manifest file from your website to their iOS device. The manifest file instructs the device to download and install the apps referenced in the manifest file.

You can distribute the URL for downloading the manifest file by SMS or email, or by embedding it in another enterprise app you create.

It's up to you to design and host the website used to distribute apps. Make sure that users are authenticated, perhaps using basic auth or directory-based authentication, and that the website is accessible from your intranet or the Internet. You can place the app and manifest file in a hidden directory, or in any location that's readable using HTTPS.

If you create a self-service portal, consider adding a web clip to the user's Home screen so it's easy to direct them back to the portal for future deployment information, such as new configuration profiles, recommended App Store apps, and enrollment in a mobile device management solution.

Prepare an in-house app for wireless distribution

To prepare your in-house app for wireless distribution, you build an archived version (an .ipa file), and a manifest file that enables wireless distribution and installation of the app.

You use Xcode to create an app archive. Sign the app using your distribution certificate and include your enterprise deployment provisioning profile in the archive. For more information about building and archiving apps, visit the [iOS Dev Center](#) or refer to the *Xcode User Guide*, available from the Help menu in Xcode.

About the wireless manifest file

The manifest file is an XML plist. It's used by an iOS device to find, download, and install apps from your web server. The manifest file is created by Xcode, using information you provide when you share an archived app for enterprise distribution.

The following fields are required:

- *URL*: The fully qualified HTTPS URL of the app (.ipa) file.
- *display-image*: A 57-by-57-pixel PNG image that's displayed during download and installation. Specify the image's fully qualified URL.
- *full-size-image*: A 512-by-512-pixel PNG image that represents the app in iTunes.
- *bundle-identifier*: Your app's bundle identifier, exactly as specified in your Xcode project.
- *bundle-version*: Your app's bundle version, as specified in your Xcode project.
- *title*: The name of the app, which is displayed during download and installation.

For Newsstand apps only, the following fields are required:

- *newsstand-image*: A full-size PNG image for display on the Newsstand shelf.
- *UINewsstandBindingEdge* and *UINewsstandBindingType*: These keys must match those in your Newsstand app's info.plist.
- *UINewsstandApp*: Indicates that the app is a Newsstand app.

Optional keys you can use are described in the sample manifest file. For example, you can use the MD5 keys if your app file is large and you want to ensure download integrity beyond the error checking normally done for TCP communications.

You can install more than one app with a single manifest file, by specifying additional members of the items array.

A sample manifest file is included at the end of this appendix.

Construct your website

Upload these items to an area of your website that your authenticated users can access:

- The app (.ipa) file
- The manifest (.plist) file

Your website can be a single page that links to the manifest file. When a user taps a web link, the manifest file is downloaded, which triggers the downloading and installation of the apps it describes.

Here's a sample link:

```
<a href="itms-services://?action=download-  
manifest&url=https://example.com/manifest.  
plist">Install App</a>
```

Don't add a web link to the archived app (.ipa). The .ipa file is downloaded by the device when the manifest file is loaded. Although the protocol portion of the URL is itms-services, the iTunes Store isn't involved in this process.

Also make sure your .ipa file is accessible over HTTPS and that your site is signed with a certificate that's trusted by iOS. Installation fails if a self-signed certificate doesn't have a trusted anchor and can't be validated by the iOS device.

Set server MIME types

You may need to configure your web server so the manifest file and app file are transmitted correctly.

For OS X Server, add the following MIME types to the web service's MIME Types settings:

```
application/octet-stream ipa
text/xml plist
```

For IIS, use IIS Manager to add the MIME type in the Properties page of the server:

```
.ipa application/octet-stream
.plist text/xml
```

Troubleshoot wireless app distribution

If wireless app distribution fails with an “unable to download” message:

- Make sure the app is signed correctly. Test it by installing it on a device using Apple Configurator, and see if any errors occur.
- Make sure the link to the manifest file is correct and the manifest file is accessible to web users.
- Make sure the URL to the .ipa file (in the manifest file) is correct and the .ipa file is accessible to web users over HTTPS.

Network configuration requirements

If the devices are connected to a closed internal network, you should let iOS devices access the following:

- *ax.init.itunes.apple.com*: The device obtains the current file-size limit for downloading apps over the cellular network. If this website isn't reachable, installation may fail.
- *ocsp.apple.com*: The device contacts this website to check the status of the distribution certificate used to sign the provisioning profile.

Provide updated apps

Apps you distribute yourself aren't automatically updated. When you have a new version for users to install, notify them of the update and instruct them to install the app. Consider having the app check for updates and notify the user when it opens. If you're using wireless app distribution, the notification can provide a link to the manifest file of the updated app.

If you want users to keep the app's data stored on their device, make sure the new version uses the same bundle identifier as the one it's replacing, and tell users not to delete their old version before installing the new one. The new version replaces the old one and keeps data stored on the device, if the bundle identifiers match.

Distribution provisioning profiles expire 12 months after they're issued. After the expiration date, the profile is removed and the app won't launch.

Before a provisioning profile expires, go to the [iOS Dev Center](#) website to create a new profile for the app. Create a new app archive (.ipa) with the new provisioning profile, for users installing the app for the first time.

If users already have the app, you may want to time your next released version so that it includes the new provisioning profile. If not, you can distribute just the new .mobileprovision file, so users won't have to install the app again. The new provisioning profile overrides the one already in the app archive.

Provisioning profiles can be installed and managed using MDM and then downloaded and installed by users through an app update or using MDM.

If your distribution certificate expires, the app won't launch. Your distribution certificate is valid for three years from when it was issued, or until your Enterprise Developer Program membership expires, whichever comes first. To keep your certificate from expiring, be sure to renew your membership before it expires.

You can have two distribution certificates active at the same time, with each independent from the other. The second certificate provides an overlapping period in which you can update your apps before the first certificate expires. When you request your second distribution certificate from the iOS Dev Center, be sure not to revoke your first certificate.

Certificate validation

The first time a user opens an app, the distribution certificate is validated by contacting Apple's OCSP server. If the certificate has been revoked, the app won't launch. Inability to contact or get a response from the OCSP server isn't interpreted as a revocation. To verify the status, the device must be able to reach ocsp.apple.com. See Network configuration requirements.

The OCSP response is cached on the device for the period of time specified by the OCSP server—currently, between three and seven days. The validity of the certificate isn't checked again until the device has restarted and the cached response has expired. If a revocation is received at that time, the app won't launch.

Revoking a distribution certificate invalidates all of the apps you've signed with it. Revoke a certificate only as a last resort—that is, if you're sure the private key is lost or you think the certificate has been compromised.

Sample app manifest file

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <!-- array of downloads. -->
  <key>items</key>
  <array>
    <dict>
      <!-- an array of assets to download -->
      <key>assets</key>
      <array>
        <!-- software-package: the ipa to install. -->
        <dict>
          <!-- required. the asset kind. -->
          <key>kind</key>
          <string>software-package</string>
```

```

        <!-- optional. md5 every n bytes. will restart a chunk if md5
fails. -->
        <key>md5-size</key>
        <integer>10485760</integer>
        <!-- optional. array of md5 hashes for each "md5-size" sized
chunk. -->
        <key>md5s</key>
        <array>
            <string>41fa64bb7a7cae5a46bfb45821ac8bba</string>
            <string>51fa64bb7a7cae5a46bfb45821ac8bba</string>
        </array>
        <!-- required. the URL of the file to download. -->
        <key>url</key>
        <string>https://www.example.com/apps/foo.ipa</string>
    </dict>
    <!-- display-image: the icon to display during download.-->
    <dict>
        <key>kind</key>
        <string>display-image</string>
        <!-- optional. indicates if icon needs shine effect applied. -->
        <key>needs-shine</key>
        <true/>
        <key>url</key>
        <string>https://www.example.com/image.57x57.png</string>
    </dict>
    <!-- full-size-image: the large 512x512 icon used by iTunes. -->
    <dict>
        <key>kind</key>
        <string>full-size-image</string>
        <!-- optional. one md5 hash for the entire file. -->
        <key>md5</key>
        <string>61fa64bb7a7cae5a46bfb45821ac8bba</string>
        <key>needs-shine</key>
        <true/>
        <key>url</key><string>https://www.example.com/image.512x512.
jpg</string>
    </dict>
</array><key>metadata</key>
<dict>
    <!-- required -->
    <key>bundle-identifier</key>
    <string>com.example.fooapp</string>
    <!-- optional (software only) -->
    <key>bundle-version</key>
    <string>1.0</string>

```

```

        <!-- required. the download kind. -->
        <key>kind</key>
        <string>software</string>
        <!-- optional. displayed during download; typically company name
-->
        <key>subtitle</key>
        <string>Apple</string>
        <!-- required. the title to display during the download. -->
        <key>title</key>
        <string>Example Corporate App</string>
    </dict>
</dict>
</array>
</dict>
</plist>

```

For more information about profile keys and attributes, see [Configuration Profile Key Reference](#).