iPhone and iPad in Business
Deployment Scenarios

October 2011

Learn how iPhone and iPad integrate seamlessly into enterprise environments with these deployment scenarios.

- Microsoft Exchange ActiveSync
- Standards-Based Services
- Virtual Private Networks
- Wi-Fi
- Digital Certificates
- Security Overview
- Mobile Device Management

Deploying iPhone and iPad
Exchange ActiveSync

Supported Exchange ActiveSync security policies

- Remote wipe
- Enforce password on device
- Minimum password length
- Maximum failed password attempts (before local wipe)
- Require both numbers and letters
- Inactivity time in minutes (1 to 60 minutes)

Additional Exchange ActiveSync policies (for Exchange 2007 and 2010 only)

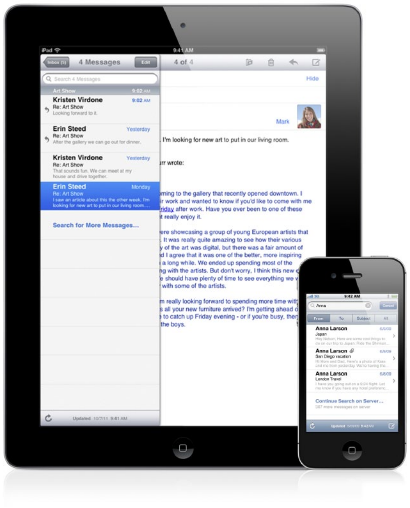
- Allow or prohibit simple password
- Password expiration
- Password history
- Policy refresh interval
- Minimum number of complex characters in password
- Require manual syncing while roaming
- Allow camera
- Allow web browsing

iPhone and iPad can communicate directly with your Microsoft Exchange Server via Microsoft Exchange ActiveSync (EAS), enabling push email, calendar, contacts, 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 company currently enables Exchange ActiveSync, you have the necessary services in place to support iPhone and iPad—no additional configuration is required. If you have Exchange Server 2003, 2007, or 2010 but your company is new to Exchange ActiveSync, review the following steps.

Exchange ActiveSync Setup

Network configuration overview

- Check to ensure port 443 is open on the firewall. If your company allows Outlook Web Access, port 443 is most likely already open.
- On the Front-End Server, verify that a server certificate is installed and enable SSL for the Exchange ActiveSync virtual directory in IIS.
- If you're using a Microsoft Internet Security and Acceleration (ISA) Server, verify that a server certificate is installed and update the public DNS to resolve incoming connections.
- Make sure the DNS for your network returns a single, externally routable address to the Exchange ActiveSync server for both intranet and Internet clients. This is required so the device can use the same IP address for communicating with the server when both types of connections are active.
- If you're using a Microsoft ISA Server, create a web listener as well as an Exchange web client access publishing rule. See Microsoft's documentation for details.
- For all firewalls and network appliances, set the Idle Session Timeout to 30 minutes. For information about heartbeat and timeout intervals, refer to the Microsoft Exchange documentation at http://technet.microsoft.com/en-us/library/cc182270.aspx.
- Configure mobile features, policies, and device security settings using the Exchange System Manager. For Exchange Server 2007 and 2010, this is done in the Exchange Management Console.
- Download and install the Microsoft Exchange ActiveSync Mobile Administration Web Tool, which is necessary to initiate a remote wipe. For Exchange Server 2007 and 2010, remote wipe can also be initiated using Outlook Web Access or the Exchange Management Console.



Other Exchange ActiveSync services

- Global Address List lookup
- Accept and create calendar invitations
- Sync tasks
- Flag email messages
- Sync Reply and Forward flags with Exchange Server 2010
- Mail search on Exchange Server 2007 and 2010
- Support for multiple Exchange ActiveSync accounts
- Certificate-based authentication
- Email push to selected folders
- Autodiscover

Basic authentication (username and password)

- Enable Exchange ActiveSync for specific users or groups using the Active Directory service. These are enabled by default for all mobile devices at the organizational level in Exchange Server 2003, 2007, and 2010. For Exchange Server 2007 and 2010, see Recipient Configuration in the Exchange Management Console.
- By default, Exchange ActiveSync is configured for basic user authentication. It's recommended that you enable SSL for basic authentication to ensure credentials are encrypted during authentication.

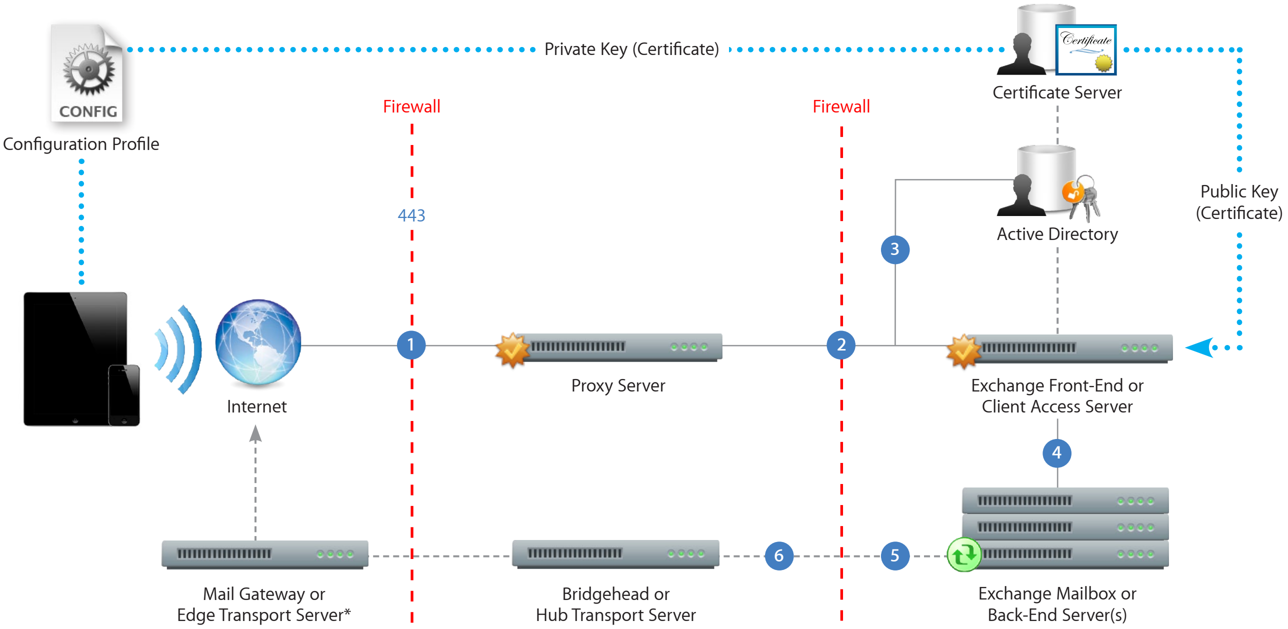
Certificate-based authentication

- Install enterprise certificate services on a member server or domain controller in your domain (this will be your certificate authority server).
- Configure IIS on your Exchange front-end server or Client Access Server to accept certificate-based authentication for the Exchange ActiveSync virtual directory.
- To allow or require certificates for all users, turn off "Basic authentication" and select either "Accept client certificates" or "Require client certificates."
- Generate client certificates using your certificate authority server. Export the public key and configure IIS to use this key. Export the private key and use a Configuration Profile to deliver this key to iPhone and iPad. Certificate-based authentication can only be configured using a Configuration Profile.

For more information on certificate services, please refer to resources available from Microsoft.

Exchange ActiveSync Deployment Scenario

This example shows how iPhone and iPad connect to a typical Microsoft Exchange Server 2003, 2007, or 2010 deployment.



*Depending on your network configuration, the Mail Gateway or Edge Transport Server may reside within the perimeter network (DMZ).

1. iPhone and iPad request access to Exchange ActiveSync services over port 443 (HTTPS). (This is the same port used for Outlook Web Access and other secure web services, so in many deployments this port is already open and configured to allow SSL encrypted HTTPS traffic.)

2. ISA provides access to the Exchange Front-End or Client Access Server. ISA is configured as a proxy, or in many cases a reverse proxy, to route traffic to the Exchange Server.

3. Exchange Server authenticates the incoming user via the Active Directory service and the certificate server (if using certificate-based authentication).

4. If the user provides the proper credentials and has access to Exchange ActiveSync services, the Front-End Server establishes a connection to the appropriate mailbox on the Back-End Server (via the Active Directory Global Catalog).

5. The Exchange ActiveSync connection is established. Updates/changes are pushed over the air, and any changes made on iPhone and iPad are reflected on the Exchange Server.

6. Sent mail items are also synchronized with the Exchange Server via Exchange ActiveSync (step 5). To route outbound email to external recipients, mail is typically sent through a Bridgehead (or Hub Transport) Server to an external Mail Gateway (or Edge Transport Server) via SMTP. Depending on your network configuration, the external Mail Gateway or Edge Transport Server could reside within the perimeter network or outside the firewall.

Deploying iPhone and iPad
Standards-Based Services

Enter your account information

IMAP | POP

Name: John Appleseed
Address: jappleseed@acme.com
Description: IMAP Account

Incoming Mail Server
Host Name: mail.acme.com
User Name: jappleseed
Password: ●●●●●●●●●

Outgoing Mail Server
Host Name: smtp.acme.com
User Name: jappleseed
Password: ●●●●●●●●●

New Account

IMAP | POP

Name: John Appleseed
Address: jappleseed@acme.com
Description: ACME Mail

Incoming Mail Server
Host Name: mail.acme.com
User Name: jappleseed
Password: ●●●●●●●●

With support for the IMAP mail protocol, LDAP directory services, and CalDAV calendaring and CardDAV contacts protocols, iOS can integrate with just about any standards-based mail, calendar, and contacts environment. And if your network environment is configured to require user authentication and SSL, iPhone and iPad provide a secure approach to accessing standards-based corporate email, calendar, tasks, and contacts.

In a typical deployment, iPhone and iPad establish direct access to IMAP and SMTP mail servers to receive and send email over the air, and can also wirelessly sync notes with IMAP-based servers. iOS devices can connect to your company's LDAPv3 corporate directories, giving users access to corporate contacts in the Mail, Contacts, and Messages applications. Synchronization with your CalDAV server allows users to wirelessly create and accept calendar invitations, receive calendar updates, and sync tasks with the Reminders app. And CardDAV support allows your users to maintain a set of contacts synced with your CardDAV server using the vCard format. All network servers can be located within a DMZ subnetwork, behind a corporate firewall, or both. With SSL, iOS supports 128-bit encryption and X.509 root certificates issued by the major certificate authorities.

Network Setup

Your IT or network administrator will need to complete these key steps to enable access from iPhone and iPad to IMAP, LDAP, CalDAV, and CardDAV services:

- Open the appropriate ports on the firewall. Common ports include 993 for IMAP mail, 587 for SMTP mail, 636 for LDAP directory services, 8443 for CalDAV calendaring, and 8843 for CardDAV contacts. It's also recommended that communication between your proxy server and your back-end IMAP, LDAP, CalDAV, and CardDAV servers be set to use SSL and that digital certificates on your network servers be signed by a trusted certificate authority (CA) such as VeriSign. This important step ensures that iPhone and iPad recognize your proxy server as a trusted entity within your corporate infrastructure.
- For outbound SMTP email, port 587, 465, or 25 must be opened to allow email to be sent. iOS automatically checks for port 587, then 465, and then 25. Port 587 is the most reliable, secure port because it requires user authentication. Port 25 does not require authentication, and some ISPs block this port by default to prevent spam.

Common ports

- IMAP/SSL: 993
- SMTP/SSL: 587
- LDAP/SSL: 636
- CalDAV/SSL: 8443, 443
- CardDAV/SSL: 8843, 443

IMAP or POP-enabled mail solutions

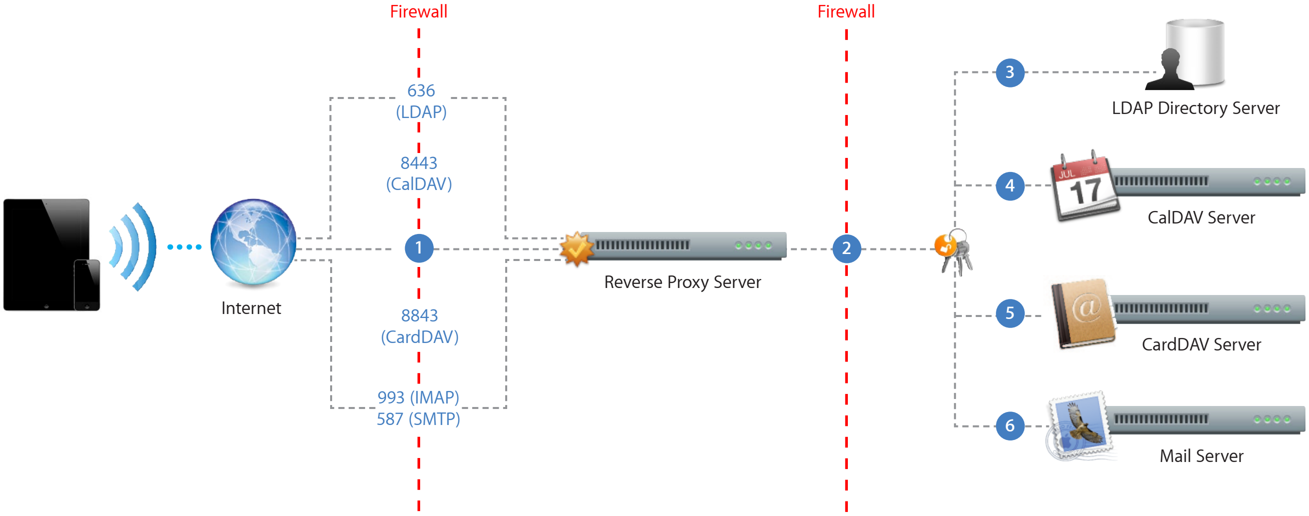
iOS supports industry-standard IMAP4- and POP3-enabled mail servers on a range of server platforms, including Windows, UNIX, Linux, and Mac OS X.

CalDAV and CardDAV standards

iOS supports the CalDAV calendaring and CardDAV contacts protocols. Both protocols have been standardized by the IETF. More information can be found through the CalConnect consortium at http://caldav.calconnect.org/ and http://carddav.calconnect.org/.

Deployment Scenario

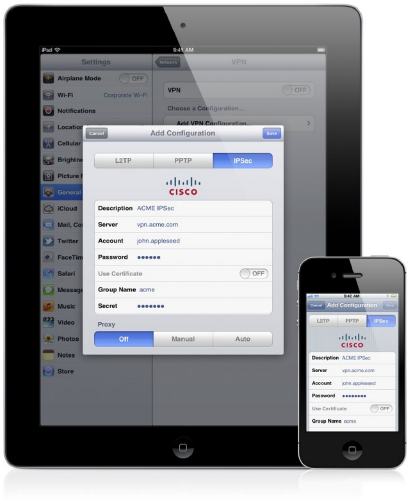
This example shows how iPhone and iPad connect to a typical IMAP, LDAP, CalDAV, and CardDAV deployment.



1. iPhone and iPad request access to network services over the designated ports.
2. Depending on the service, users must authenticate either with the reverse proxy or directly with the server to obtain access to corporate data. In all cases, connections are relayed by the reverse proxy, which functions as a secure gateway, typically behind the company's Internet firewall. Once authenticated, users can access their corporate data on the back-end servers.
3. iPhone and iPad provide lookup services on LDAP directories, giving users the ability to search for contacts and other address book information on the LDAP server.
4. For CalDAV calendars, users can access and update calendars.
5. CardDAV contacts are stored on the server and can also be accessed locally on iPhone and iPad. Changes to fields in CardDAV contacts are synced back to the CardDAV server.
6. For IMAP mail services, existing and new messages can be read on iPhone and iPad through the proxy connection with the mail server. Outgoing mail is sent to the SMTP server, with copies placed in the user's Sent folder.

October 2011 L419827B

Deploying iPhone and iPad
Virtual Private Networks



Description ACME IPSec
Server vpn.acme.com
Account john.appleseed
Password ••••••
Use Certificate OFF
Group Name acme
Secret •••••••

Secure access to private corporate networks is available on iPhone and iPad using established industry-standard virtual private network (VPN) protocols. Users can easily connect to enterprise systems via the built-in VPN client in iOS or through third-party applications from Juniper, Cisco, and F5 Networks.

Out of the box, iOS supports Cisco IPSec, L2TP over IPSec, and PPTP. If your organization supports one of these protocols, no additional network configuration or third-party applications are required to connect iPhone and iPad to your VPN.

Additionally, iOS supports SSL VPN, enabling access to Juniper SA Series, Cisco ASA, and F5 BIG-IP Edge Gateway SSL VPN servers. Users simply download a VPN client application developed by Juniper, Cisco, or F5 from the App Store to get started. Like other VPN protocols supported in iOS, SSL VPN can be configured manually on the device or via Configuration Profile.

iOS supports industry-standard technologies such as IPv6, proxy servers, and split-tunneling, providing a rich VPN experience when connecting to corporate networks. And iOS works with a variety of authentication methods including password, two-factor token, and digital certificates. To streamline the connection in environments where certificate-based authentication is used, iOS features VPN On Demand, which dynamically initiates a VPN session when connecting to specified domains.

Supported Protocols and Authentication Methods

SSL VPN
Supports user authentication by password, two-factor token, and certificates.

Cisco IPSec
Supports user authentication by password, two-factor token, and machine authentication by shared secret and certificates.

L2TP over IPSec
Supports user authentication by MS-CHAP v2 Password, two-factor token, and machine authentication by shared secret.

PPTP
Supports user authentication by MS-CHAP v2 Password and two-factor token.

VPN On Demand

For configurations using certificate-based authentication, iOS supports VPN On Demand. VPN On Demand will establish a connection automatically when accessing predefined domains, providing a seamless VPN connectivity experience for users.

This is a feature of iOS that does not require additional server configuration. The configuration of VPN On Demand takes place via a Configuration Profile or can be configured manually on the device.

The VPN On Demand options are:

Always
Initiates a VPN connection for any address that matches the specified domain.

Never
Does not initiate a VPN connection for addresses that match the specified domain, but if VPN is already active, it may be used.

Establish if needed
Initiates a VPN connection for addresses that match the specified domain only after a DNS look-up has failed.

VPN Setup

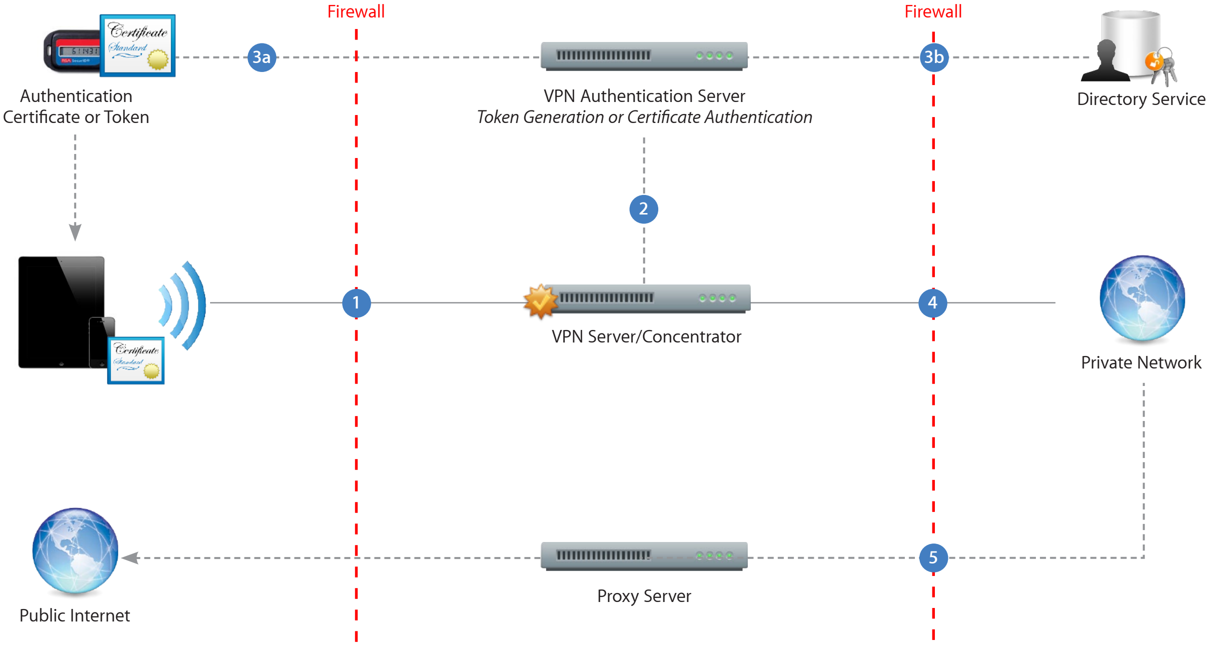
- iOS integrates with many existing VPN networks, with minimal configuration necessary. The best way to prepare for deployment is to check whether iOS supports your company's existing VPN protocols and authentication methods.
- It's recommended that you review the authentication path to your authentication server to make sure standards supported by iOS are enabled within your implementation.
- If you plan to use certificate-based authentication, ensure you have your public key infrastructure configured to support device- and user-based certificates with the corresponding key distribution process.
- If you want to configure URL-specific proxy settings, place a PAC file on a web server that is accessible with the basic VPN settings and ensure that it is hosted with the application/x-ns-proxy-autoconfig MIME type.

Proxy Setup

For all configurations, you can also 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 file. For WPAD, iPhone and iPad will query DHCP and DNS for the appropriate settings.

Deployment Scenario

The example depicts a typical deployment with a VPN server/concentrator as well as an authentication server controlling access to enterprise network services.



1. iPhone and iPad request access to network services.
2. The VPN server/concentrator receives the request and then passes it to the authentication server.
3. In a two-factor token environment, the authentication server would then manage a time-synchronized token key generation with the key server. If a certificate authentication method is deployed, an identity certificate needs to be distributed prior to authentication. If a password method is deployed, the authentication process proceeds with user validation.
4. Once a user is authenticated, the authentication server validates user and group policies.
5. After user and group policies are validated, the VPN server provides tunneled and encrypted access to network services.
6. If a proxy server is in use, iPhone and iPad connect through the proxy server for access to information outside the firewall.

October 2011 L419828B

Deploying iPhone and iPad
Wi-Fi

Airplane Mode OFF
Wi-Fi Corporate Wi-Fi
VPN OFF
Notifications
Location Services On
Cellular Data
Brightness & Wallpaper
Picture Frame
General
iCloud
Mail, Contacts, Calendars
Twitter
FaceTime
Safari
Messages
Music
Video
Photos
Notes
Store
AirStrip

Wi-Fi ON
Choose a Network…
✓ Corporate Wi-Fi
Guest Access Wi-Fi
Other…
Ask to Join Networks OFF

Known networks will be joined automatically. If no known networks are available, you will have to manually select a network.

Wi-Fi ON
Choose a Network…
✓ Corporate Wi-Fi
Guest Access Wi-Fi
Other…
Ask to Join Networks OFF

Known networks will be joined automatically. If no known networks are available, you will have to manually select a network.

Wireless security protocols

- WEP
- WPA Personal
- WPA Enterprise
- WPA2 Personal
- WPA2 Enterprise

802.1X authentication methods

- EAP-TLS
- EAP-TTLS
- EAP-FAST
- EAP-SIM
- PEAPv0 (EAP-MS-CHAP v2)
- PEAPv1 (EAP-GTC)
- LEAP

Out of the box, iPhone and iPad can securely connect to corporate or guest Wi-Fi networks, making it quick and simple to join available wireless networks whether you're on campus or on the road.

iOS supports industry-standard wireless network protocols, including WPA2 Enterprise, ensuring corporate wireless networks can be configured quickly and accessed securely. WPA2 Enterprise uses 128-bit AES encryption, a proven, block-based encryption method, providing users with the highest level of assurance that their data will remain protected.

With support for 802.1X, iOS can be integrated into a broad range of RADIUS authentication environments. 802.1X wireless authentication methods supported on iPhone and iPad include EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0, PEAPv1, and LEAP.

Users can set iPhone and iPad 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 applications such as Mail. And low-power, persistent Wi-Fi connectivity allows applications to use Wi-Fi networks to deliver push notifications.

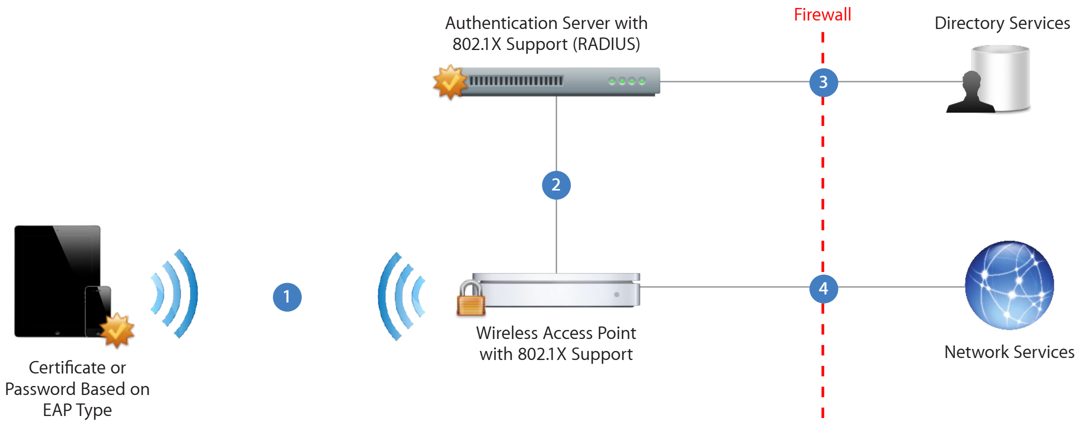
For quick setup and deployment, wireless network, security, proxy, and authentication settings can be configured using Configuration Profiles.

WPA2 Enterprise Setup

- Verify network appliances for compatibility and select an authentication type (EAP type) supported by iOS.
- Check that 802.1X is enabled on the authentication server and, if necessary, install a server certificate and assign network access permissions to users and groups.
- Configure wireless access points for 802.1X authentication and enter the corresponding RADIUS server information.
- If you plan to use certificate-based authentication, configure your public key infrastructure to support device- and user-based certificates with the corresponding key distribution process.
- Verify certificate format and authentication server compatibility. iOS supports PKCS#1 (.cer, .crt, .der) and PKCS#12.
- For additional documentation regarding wireless networking standards and Wi-Fi Protected Access (WPA), visit www.wi-fi.org.

WPA2 Enterprise/802.1X Deployment Scenario

This example depicts a typical secure wireless deployment that takes advantage of RADIUS-based authentication.



1. iPhone and iPad request access to the network. The connection is initiated in response to a user selecting an available wireless network, or is automatically initiated after a previously configured network is detected.
2. After the request is received by the access point, the request is passed to the RADIUS server for authentication.
3. The RADIUS server validates the user account utilizing the directory service.
4. Once the user is authenticated, the access point provides network access with policies and permissions as instructed by the RADIUS server.

Deploying iPhone and iPad
Digital Certificates

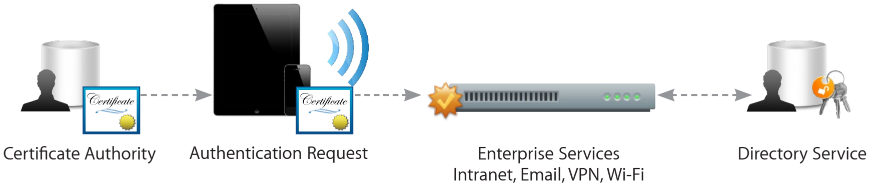
iOS supports digital certificates, giving business users secure, streamlined access to corporate services. A digital certificate is composed of a public key, information about the user, and the certificate authority that issued the certificate. Digital certificates are a form of identification that enables streamlined authentication, data integrity, and encryption.

On iPhone and iPad, certificates can be used in a variety of ways. Signing data with a digital certificate helps to ensure that information cannot be altered. Certificates can also be used to guarantee the identity of the author or "signer." Additionally, they can be used to encrypt Configuration Profiles and network communications to further protect confidential or private information.

Using Certificates in iOS

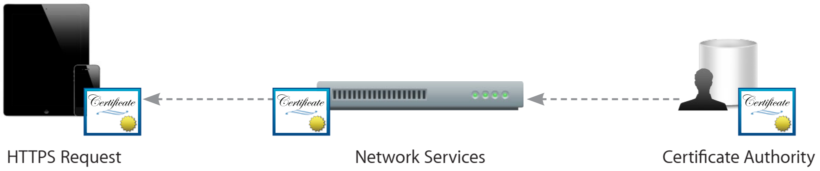
Digital certificates

Digital certificates can be used to securely authenticate users to corporate services without the need for user names, passwords, or soft tokens. In iOS, certificate-based authentication is supported for access to Microsoft Exchange ActiveSync, VPN, and Wi-Fi networks.



Server certificates

Digital certificates can also be used to validate and encrypt network communications. This provides secure communication to both internal and external websites. 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 verifies that the site's identity is legitimate and that communication with the website is protected to help prevent interception of personal or confidential data.



Supported certificate and identity formats:

- iOS supports X.509 certificates with RSA keys.
- The file extensions .cer, .crt, .der, .p12, and .pfx are recognized.

Root certificates

Out of the box, iOS includes a number of preinstalled root certificates. To view a list of the preinstalled system roots, see the Apple Support article at http://support.apple.com/kb/HT4415. If you are using a root certificate that is not preinstalled, such as a self-signed root certificate created by your company, you can distribute it using one of the methods listed in the "Distributing and Installing Certificates" section of this document.

Distributing and Installing Certificates

Distributing certificates to iPhone and iPad 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 passphrase that protects it. If a certificate's authenticity cannot be verified, users will be presented with a warning before it is added to their device.

Installing certificates via Configuration Profiles

If Configuration Profiles are being used to distribute settings for corporate services such as Exchange, VPN, or Wi-Fi, certificates can be added to the profile to streamline deployment.

Installing certificates via Mail or Safari

If a certificate is sent in an email, it will appear as an attachment. Safari can be used to download certificates from a web page. You can host a certificate on a secured website and provide users with the URL where they can download the certificate onto their devices.

Installation via the Simple Certificate Enrollment Protocol (SCEP)

SCEP is designed to provide a simplified process to handle certificate distribution for large-scale deployments. This enables Over-the-Air Enrollment of digital certificates on iPhone and iPad that can then be used for authentication to corporate services, as well as enrollment with a Mobile Device Management server.

For more information on SCEP and Over-the-Air Enrollment, visit www.apple.com/iphone/business/resources.

Certificate removal and revocation

To manually remove a certificate that has been installed, choose Settings > General > Profiles. If you remove a certificate that is required for accessing an account or network, the device will no longer be able to connect to those services.

To remove certificates over the air, a Mobile Device Management server can be used. This server can view all certificates on a device and remove ones it has installed.

Additionally, the Online Certificate Status Protocol (OCSP) is supported to check the status of certificates. When an OSCP-enabled certificate is used, iOS validates it to make sure that it has not been revoked before completing the requested task.

October 2011 L419821B

Deploying iPhone and iPad
Security Overview

iOS, the operating system at the core of iPhone and iPad, is built upon layers of security. This enables iPhone and iPad 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 at rest. iOS also provides secure protection through the use of passcode policies that can be delivered and enforced over the air. And if the device falls into the wrong hands, users and IT administrators can initiate a remote wipe command to erase private information.

When considering the security of iOS for enterprise use, it's helpful to understand the following:

- **Device security:** Methods that prevent unauthorized use of the device
- **Data security:** Protecting data at rest, even when a device is lost or stolen
- **Network security:** Networking protocols and the encryption of data in transmission
- **App security:** The secure platform foundation of iOS

These capabilities work in concert to provide a secure mobile computing platform.

Device Security

Establishing strong policies for access to iPhone and iPad is critical to protecting corporate information. Device passcodes are the front line of defense against unauthorized access and can be configured and enforced over the air. iOS devices use the unique passcode established by each user to generate a strong encryption key to further protect mail and sensitive application data on the device. Additionally, iOS provides secure methods to configure the device in an enterprise 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.

Device security

- Strong passcodes
- Passcode expiration
- Passcode reuse history
- Maximum failed attempts
- Over-the-air passcode enforcement
- Progressive passcode timeout

Passcode policies

A device passcode prevents unauthorized users from accessing data or otherwise gaining access to the device. iOS allows you to select from an extensive set of passcode requirements to meet your security needs, including timeout periods, passcode strength, and how often the passcode must be changed.

The following passcode policies are supported:

- Require passcode on device
- Allow simple value
- 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

Supported configurable policies and restrictions:

Device functionality

- Allow installing apps
- Allow Siri
- Allow use of camera
- Allow FaceTime
- Allow screen capture
- Allow automatic syncing while roaming
- Allow voice dialing
- Allow In-App Purchase
- Require store password for all purchases
- Allow multiplayer gaming
- Allow adding Game Center friends

Applications

- Allow use of YouTube
- Allow use of iTunes Store
- Allow use of Safari
- Set Safari security preferences

iCloud

- Allow backup
- Allow document sync and key-value sync
- Allow Photo Stream

Security and privacy

- Allow diagnostic data to be sent to Apple
- Allow user to accept untrusted certificates
- Force encrypted backups

Content ratings

- Allow explicit music and podcasts
- Set ratings region
- Set allowed content ratings

Policy enforcement

The policies described previously can be set on iPhone and iPad in a number of ways. Policies can be distributed as part of a Configuration Profile for users to install. A profile can be defined so that deleting the profile is only possible with an administrative password, or you can define the profile so that it is locked to the device and cannot be removed without completely erasing all of the device contents. Additionally, passcode settings can be configured remotely using Mobile Device Management (MDM) solutions that can push policies directly to the device. This enables policies to be enforced and updated without any action by the user.

Alternatively, if the device is configured to access a Microsoft Exchange account, Exchange ActiveSync policies are pushed to the device over the air. Keep in mind that the available set of policies will vary depending on the version of Exchange (2003, 2007, or 2010). Refer to *Exchange ActiveSync and iOS Devices* for a breakdown of which policies are supported for your specific configuration.

Secure device configuration

Configuration Profiles are XML files that contain device security policies and restrictions, VPN configuration information, Wi-Fi settings, email and calendar accounts, and authentication credentials that permit iPhone and iPad to work with your enterprise systems. The ability to establish passcode policies along with device settings in a Configuration Profile ensures that devices within your enterprise are configured correctly and according to security standards set by your organization. And, because Configuration Profiles can be encrypted and locked, the settings cannot 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 for which it was created. 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 via USB using the Configuration Utility or wirelessly via Over-the-Air Enrollment. In addition to these methods, subsequent encrypted Configuration Profiles can be delivered via email attachment, hosted on a website accessible to your users, or pushed to the device using MDM solutions.

Device restrictions

Device restrictions determine which features your users can access on the device. Typically, these involve network-enabled applications such as Safari, YouTube, or the iTunes Music Store, but restrictions can also control device functionality such as application installation or use of camera. Restrictions let you configure the device to meet your requirements, while permitting users to utilize the device in ways that are consistent with your business practices. Restrictions can be manually configured on each device, enforced using a Configuration Profile, or established remotely with MDM solutions. Additionally, like passcode policies, camera or web-browsing restrictions can be enforced over the air via Microsoft Exchange Server 2007 and 2010.

In addition to setting restrictions and policies on the device, the iTunes desktop application can be configured and controlled by IT. This includes disabling access to explicit content, defining which network services users can access within iTunes, and determining whether new software updates are available for users to install. For more information, refer to *Deploying iTunes for iOS Devices*.

Data Security

Data security

- Hardware encryption
- Data protection
- Remote wipe
- Local wipe
- Encrypted Configuration Profiles
- Encrypted iTunes backups

Protecting data stored on iPhone and iPad is important for any environment with sensitive corporate or customer information. In addition to encrypting data in transmission, iPhone and iPad provide hardware encryption for all data stored on the device, and additional encryption of email and application data with enhanced data protection.

If a device is lost or stolen, it's important to deactivate and erase the device. It's also a good idea to have a policy in place that will wipe the device after a defined number of failed passcode attempts, a key deterrent against attempts to gain unauthorized access to the device.

Encryption

iPhone and iPad offer hardware-based encryption. Hardware encryption uses 256-bit AES to protect all data on the device. Encryption is always enabled, and cannot be disabled by users.

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.

iOS supports S/MIME in mail, enabling iPhone and iPad to view and send encrypted email messages. Restrictions can also be used to prevent mail messages from being moved between accounts or messages received in one account being forwarded from another.

Data protection

Building on the hardware encryption capabilities of iPhone and iPad, email messages and attachments stored on the device can be further secured by using data protection features built into iOS. Data protection leverages each user's unique device passcode in concert with the hardware encryption on iPhone and iPad to generate a strong encryption key. This key prevents data from being accessed when the device is locked, ensuring that critical information is secured even if the device is compromised.

To turn on the data protection feature, simply establish a passcode on the device. The effectiveness of data protection is dependent on a strong passcode, so it is important to require and enforce a passcode stronger than four digits when establishing your corporate passcode policies. Users can verify 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.

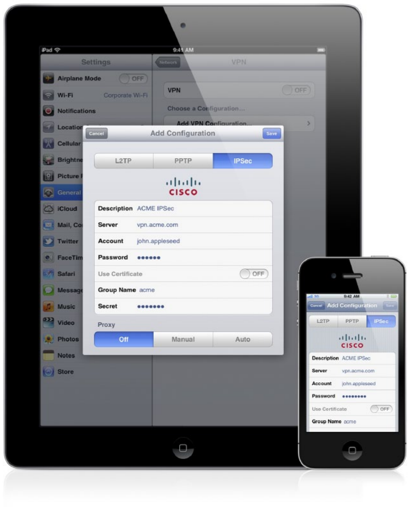
These data protection APIs are also available to developers, and can be used to secure enterprise in-house or commercial application data.

Progressive passcode timeout

iPhone and iPad can be configured to automatically initiate a wipe after several failed passcode attempts. If a user repeatedly enters the wrong passcode, iOS will be disabled for increasingly longer intervals. After too many unsuccessful attempts, all data and settings on the device will be erased.

Remote wipe

iOS supports remote wipe. If a device is lost or stolen, the administrator or device owner can issue a remote wipe command that removes all data and deactivates the device. 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 also initiate remote wipe commands directly using Outlook Web Access. Remote wipe commands can also be initiated by MDM solutions even if Exchange corporate services are not in use.



Description ACME IPSec
Server vpn.acme.com
Account john.appleseed
Password ●●●●●●
Use Certificate OFF
Group Name acme
Secret ●●●●●●●

Local wipe

Devices can also be configured 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 have the ability to enable local wipe directly within the settings. By default, iOS will automatically wipe the device after 10 failed passcode attempts. As with other passcode policies, the maximum number of failed attempts can be established via a Configuration Profile, set by an MDM server, or enforced over the air via Microsoft Exchange ActiveSync policies.

iCloud

iCloud stores music, photos, apps, calendars, documents, and more, and automatically pushes them to all of a user's devices. iCloud also backs up information, including device settings, app data, and text and MMS messages, daily over Wi-Fi. iCloud secures your content by encrypting it when sent over the Internet, storing it in an encrypted format, and using secure tokens for authentication. Additionally, iCloud features, including Photo Stream, Document Sync, and Backup, can be disabled via a Configuration Profile. For more information on iCloud security and privacy, visit http://support.apple.com/kb/HT4865.

Network Security

Mobile users must be able to access corporate information 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. iOS provides proven technologies to accomplish these security objectives for both Wi-Fi and cellular data network connections.

In addition to your existing infrastructure, each FaceTime session and iMessage conversation is encrypted end to end. iOS creates a unique ID for each user, ensuring communications are encrypted, routed, and connected properly.

VPN

Many enterprise environments have some form of virtual private network (VPN) established. These secure network services are already deployed and typically require minimal setup and configuration to work with iPhone and iPad.

Out of the box, iOS integrates with a broad range of commonly used VPN technologies through support for Cisco IPSec, L2TP, and PPTP. iOS supports SSL VPN through applications from Juniper, Cisco, and F5 Networks. Support for these protocols ensures the highest level of IP-based encryption for transmission of sensitive information.

In addition to enabling secure access to existing VPN environments, iOS offers proven methods for user authentication. Authentication via standard X.509 digital certificates provides users with streamlined access to company resources and a viable alternative to using hardware-based tokens. Additionally, certificate authentication enables iOS to take advantage of VPN On Demand, making the VPN authentication process transparent while still providing strong, credentialed access to network services. For enterprise environments in which a two-factor token is a requirement, iOS integrates with RSA SecurID and CRYPTOCard.

iOS supports network proxy configuration as well as split IP tunneling so that traffic to public or private network domains is relayed according to your specific company policies.

Network security

- Built-in Cisco IPSec, L2TP, PPTP VPN
- 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 protocols

- Cisco IPSec
- L2TP/IPSec
- PPTP
- SSL VPN

Authentication methods

- Password (MSCHAPv2)
- RSA SecurID
- CRYPTOCard
- X.509 digital certificates
- Shared secret

802.1X authentication protocols

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

Supported certificate formats

iOS supports X.509 certificates with RSA keys. The file extensions .cer, .crt, and .der are recognized.

SSL/TLS

iOS supports SSL v3 as well as Transport Layer Security (TLS v1.0, 1.1, and 1.2), the next-generation security standard for the Internet. Safari, Calendar, Mail, and other Internet applications automatically start these mechanisms to enable an encrypted communication channel between iOS and corporate services.

WPA/WPA2

iOS supports WPA2 Enterprise to provide authenticated access to your enterprise wireless network. WPA2 Enterprise uses 128-bit AES encryption, giving users the highest level of assurance that their data will remain protected when they send and receive communications over a Wi-Fi network connection. And with support for 802.1X, iPhone and iPad can be integrated into a broad range of RADIUS authentication environments.

App Security

App security

- Runtime protection
- Mandatory code signing
- Keychain services
- CommonCrypto APIs
- Application data protection

iOS is designed with security at its core. It includes a "sandboxed" approach to application runtime protection and requires application signing to ensure that applications cannot be tampered with. iOS also has a secure framework that facilitates secure storage of application and network service credentials in an encrypted keychain. For developers, it offers a common crypto architecture that can be used to encrypt application data stores.

Runtime protection

Applications on the device are "sandboxed" so they cannot access data stored by other applications. In addition, system files, resources, and the kernel are shielded from the user's application space. If an application needs to access data from another application, it can only do so using the APIs and services provided by iOS. Code generation is also prevented.

Mandatory code signing

All iOS applications must be signed. The applications provided with the device are signed by Apple. Third-party applications are signed by the developer using an Apple-issued certificate. This ensures that applications haven't been tampered with or altered. Additionally, runtime checks are made to ensure that an application hasn't become untrusted since it was last used.

The use of custom or in-house applications can be controlled with a provisioning profile. Users must have the provisioning profile installed to execute the application. Provisioning profiles can be installed or revoked over the air using MDM solutions. Administrators can also restrict the use of an application to specific devices.

Secure authentication framework

iOS provides a secure, encrypted keychain for storing digital identities, user names, and passwords. Keychain data is partitioned so that credentials stored by third-party applications cannot be accessed by applications with a different identity. This provides the mechanism for securing authentication credentials on iPhone and iPad across a range of applications and services within the enterprise.

Common Crypto architecture

Application developers have access to encryption APIs that they can use to further protect their application data. Data can be symmetrically encrypted using proven methods such as AES, RC4, or 3DES. In addition, iPhone and iPad provide hardware acceleration for AES encryption and SHA1 hashing, maximizing application performance.

Application data protection

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

Managed apps

An MDM server can manage third-party apps from the App Store, as well as enterprise in-house applications. Designating an app as managed enables the server to specify whether the app and its data can be removed from the device by the MDM server. Additionally, the server can prevent managed app data from being backed up to iTunes and iCloud. This allows IT to 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 device. Managed apps require a user's acceptance before they are installed. For more information about managed apps, view the *Mobile Device Management Overview* at www.apple.com/business/mdm.

Revolutionary Devices, Security Throughout

iPhone and iPad provide encrypted protection of data in transit, at rest, and when backed up to iCloud or iTunes. Whether a user is accessing corporate email, visiting a private website, or authenticating to the corporate network, iOS provides assurance that only authorized users can access sensitive corporate information. And, with its support for enterprise-grade networking and comprehensive methods to prevent data loss, you can deploy iOS devices with confidence that you are implementing proven mobile device security and data protection.

Deploying iPhone and iPad
Mobile Device Management

Install Profile

ACME Inc. MDM
ACME Inc.
Verified

Description: Install this profile to enroll your iPad in MDM at ACME Inc. to get access to Email, W-Fi and VPN.
Signed: iPhone Configuration Utility (8545FFEC-4B2B-4201-9A5E-F12B6B35B7D0)
Received: Nov 9, 2010
Contains: Certificate
Mobile Device Management

More Details

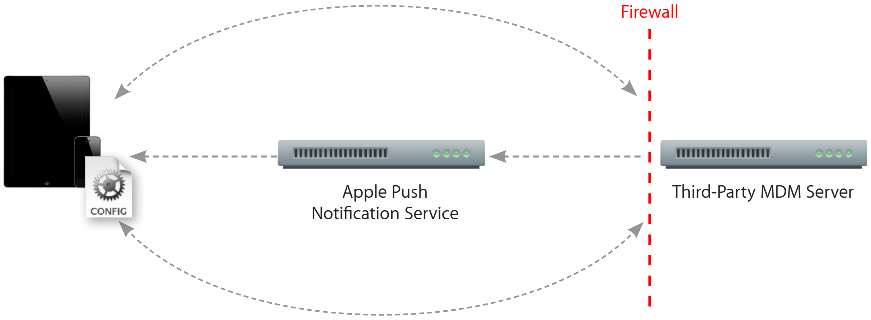
No profiles are currently installed

iOS supports Mobile Device Management (MDM), giving businesses the ability to manage scaled deployments of iPhone and iPad across their organizations. These MDM capabilities are built upon existing iOS technologies like Configuration Profiles, Over-the-Air Enrollment, and the Apple Push Notification service, and can be integrated with in-house or third-party server solutions. This gives IT departments the ability to securely enroll iPhone and iPad in an enterprise environment, wirelessly configure and update settings, monitor compliance with corporate policies, and even remotely wipe or lock managed devices.

Managing iPhone and iPad

Management of iOS devices takes place via a connection to a Mobile Device Management server. This server can be built in-house by IT or purchased from a third-party solution provider. The device communicates with the server to see if there are tasks pending and responds with the appropriate actions. These tasks can include updating policies, providing requested device or network information, or removing settings and data.

Most management functions are completed behind the scenes with no user interaction required. For example, if an IT department updates its VPN infrastructure, the MDM server can configure iPhone and iPad with new account information over the air. The next time VPN is used by the employee, the appropriate configuration is already in place, so the employee doesn't need to call the help desk or manually modify settings.



MDM and the Apple Push Notification Service

When an MDM server wants to communicate with iPhone or iPad, a silent notification is sent to the device via the Apple Push Notification service, prompting it to check in with the server. The process of notifying the device does not send any proprietary information to or from the Apple Push Notification service. The only task performed by the push notification is to wake the device so it checks in with the MDM server. All configuration information, settings, and queries are sent directly from the server to the iOS device over an encrypted SSL/TLS connection between the device and the MDM server. iOS handles all MDM requests and actions in the background to limit the impact on the user experience, including battery life, performance, and reliability.

In order for the push notification server to recognize commands from the MDM server, a certificate must first be installed on the server. This certificate must be requested and downloaded from the Apple Push Certificates Portal. Once the Apple Push Notification certificate is uploaded into the MDM server, devices can begin to be enrolled. For more information on requesting an Apple Push Notification certificate for MDM, visit www.apple.com/business/mdm.

Apple Push Notification network setup

When MDM servers and iOS devices are behind a firewall, some network configuration may need to take place in order for the MDM service to function properly. To send notifications from an MDM server to Apple Push Notification service, TCP port 2195 needs to be open. To reach the feedback service, TCP port 2196 will need to be open as well. For devices connecting to the push service over Wi-Fi, TCP port 5223 should be open.

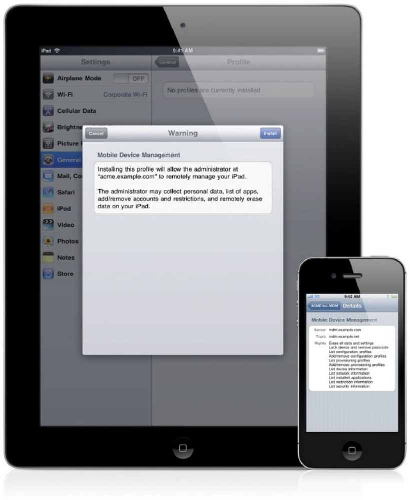
The IP address range for the push service is subject to change; the expectation is that an MDM server will connect by hostname rather than by IP address. The push service uses a load-balancing scheme that yields a different IP address for the same hostname. This hostname is gateway.push.apple.com (and gateway.sandbox.push.apple.com for the development push notification environment). Additionally, the entire 17.0.0.0/8 address block is assigned to Apple so firewall rules can be established to specify that range.

For more information, consult your MDM vendor or view *Developer Technical Note TN2265* in the iOS Developer Library at http://developer.apple.com/library/ios/#technotes/tn2265/_index.html.

Enrollment

Once the Mobile Device Management server and network are configured, the first step in managing an iPhone or iPad is to enroll it with an MDM server. This creates a relationship between the device and the server, allowing it to be managed on demand without further user interaction.

This can be done by connecting iPhone or iPad to a computer via USB, but most solutions deliver the enrollment profile wirelessly. Some MDM vendors use an app to kickstart this process, others initiate enrollment by directing users to a web portal. Each method has its benefits, and both are used to trigger the Over-the-Air Enrollment process via Safari.



Mobile Device Management

Installing this profile will allow the administrator at "acme.example.com" to remotely manage your iPad.

The administrator may collect personal data, list of apps, add/remove accounts and restrictions, and remotely erase data on your iPad.

Mobile Device Management

Server mdm.example.com
Topic mdm.example.net
Rights Erase all data and settings
Lock device and remove passcode
List configuration profiles
Add/remove configuration profiles
List provisioning profiles
Add/remove provisioning profiles
List device information
List network information
List installed applications
List restriction information
List security information

iOS and SCEP
iOS supports the Simple Certificate Enrollment Protocol (SCEP). SCEP is an Internet draft in the IETF, and is designed to provide a simplified way of handling certificate distribution for large-scale deployments. This enables over-the-air enrollment of identity certificates to iPhone and iPad that can be used for authentication to corporate services.

Enrollment process overview

The process of Over-the-Air Enrollment involves phases that are combined in an automated workflow to provide the most scalable way to securely enroll devices in an enterprise environment. These phases include:

1. User authentication

User authentication ensures that incoming enrollment requests are from authorized users and that the user's device information is captured prior to proceeding with certificate enrollment. Administrators can prompt the user to begin the process of enrollment via a web portal, email, SMS message, or even an app.

2. Certificate enrollment

After the user is authenticated, iOS generates a certificate enrollment request using the Simple Certificate Enrollment Protocol (SCEP). This enrollment request communicates directly to the enterprise Certificate Authority (CA), and enables iPhone and iPad to receive the identity certificate from the CA in response.

3. Device configuration

Once an identity certificate is installed, the device can receive encrypted configuration information over the air. This information can only be installed on the device it is intended for and contains the settings needed to connect to the MDM server.

At the end of the enrollment process, the user will be presented with an installation screen that describes what access rights the MDM server will have on the device. By agreeing to the profile installation, the user's device is automatically enrolled without further interaction.

Once iPhone and iPad are enrolled as managed devices, they can be dynamically configured with settings, queried for information, or remotely wiped by the MDM server.

Configuration

To configure a device with accounts, policies, and restrictions, the MDM server sends files known as Configuration Profiles to the device that are installed automatically. Configuration Profiles are XML files that contain settings that permit the device to work with your enterprise systems, including account information, passcode policies, restrictions, and other device settings. When combined with the previously discussed process of enrollment, device configuration provides IT with assurance that only trusted users are accessing corporate services, and that their devices are properly configured with established policies.

And because Configuration Profiles can be signed and encrypted, the settings cannot be altered or shared with others.

Supported configurable settings

Accounts

- Exchange ActiveSync
- IMAP/POP Email
- Wi-Fi
- VPN
- LDAP
- CardDAV
- CalDAV
- Subscribed calendars

Passcode policies

- Require passcode on device
- Allow simple value
- 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

Security and privacy

- Allow diagnostic data to be sent to Apple
- Allow user to accept untrusted certificates
- Force encrypted backups

Other settings

- Credentials
- Web clips
- SCEP settings
- APN settings

Device functionality

- Allow installing apps
- Allow Siri
- Allow use of camera
- Allow FaceTime
- Allow screen capture
- Allow automatic syncing while roaming
- Allow voice dialing
- Allow In-App Purchase
- Require store password for all purchases
- Allow multiplayer gaming
- Allow adding Game Center friends

Applications

- Allow use of YouTube
- Allow use of iTunes Store
- Allow use of Safari
- Set Safari security preferences

iCloud

- Allow backup
- Allow document sync and key-value sync
- Allow Photo Stream

Content ratings

- Allow explicit music and podcasts
- Set ratings region
- Set allowed content ratings

Querying Devices

In addition to configuration, an MDM server has the ability to query devices for a variety of information. This information can be used to ensure that devices continue to comply with required policies.

Supported queries

Device information

- Unique Device Identifier (UDID)
- Device name
- iOS and build version
- Model name and number
- Serial number
- Capacity and space available
- IMEI
- Modem firmware
- Battery level

Network information

- ICCID
- Bluetooth® and Wi-Fi MAC addresses
- Current carrier network
- Subscriber carrier network
- Carrier settings version
- Phone number
- Data roaming setting (on/off)

Compliance and security information

- Configuration Profiles installed
- Certificates installed with expiry dates
- List all restrictions enforced
- Hardware encryption capability
- Passcode present

Applications

- Applications installed (app ID, name, version, size, and app data size)
- Provisioning Profiles installed with expiry dates

Management

With Mobile Device Management, there are a number of functions an MDM server can perform on iOS devices. These tasks include installing and removing Configuration and Provisioning Profiles, managing apps, ending the MDM relationship, and remotely wiping a device.

Managed settings

During the initial process of configuring a device, an MDM server pushes Configuration Profiles to iPhone and iPad that are installed behind the scenes. Over time, the settings and policies put in place at the time of enrollment may need to be updated or changed. To make these changes, an MDM server can install new Configuration Profiles and modify or remove existing profiles at any time. Additionally, context-specific configurations may need to be installed on iOS devices, depending on a user's location or role in the organization. As an example, if a user is traveling internationally, an MDM server can require that mail accounts sync manually instead of automatically. An MDM server can even remotely disable voice or data services in order to prevent a user from incurring roaming fees from a wireless provider.

Managed apps

An MDM server can manage third-party apps from the App Store, as well as enterprise in-house applications. The server can remove managed apps and their associated data on demand or specify whether the apps are removed when the MDM profile is removed. Additionally, the MDM server can prevent managed app data from being backed up to iTunes and iCloud.

To install a managed app, the MDM server sends an installation command to the user's device. Managed apps require a user's acceptance before they are installed. When an MDM server requests the installation of a managed app from the App Store, the app will be redeemed with the iTunes account that is used at the time the app is installed. Paid apps will require the MDM server to send a Volume Purchasing Program (VPP) redemption code. For more information on VPP, visit www.apple.com/business/vpp/. Apps from the App Store cannot be installed on a user's device if the App Store has been disabled.

Removing or wiping devices

If a device is found to be out of policy, lost, or stolen, or if an employee leaves the company, an MDM server can take action to protect corporate information in a number of ways.

An IT administrator can end the MDM relationship with a device by removing the Configuration Profile that contains the MDM server information. In doing so, all the accounts, settings, and apps it was responsible for installing are removed. Alternatively, IT can keep the MDM Configuration Profile in place and use MDM only to remove the specific Configuration Profiles, Provisioning Profiles, and managed apps they want to delete. This approach keeps the device managed by MDM and eliminates the need to re-enroll once it is back within policy.

Both methods give IT the ability to ensure information is only available to compliant users and devices, and ensures corporate data is removed without interfering with a user's personal data such as music, photos, or personal apps.

To permanently delete all media and data on the device and restore it to factory settings, MDM can remotely wipe iPhone and iPad. If a user is still looking for the device, IT can also choose to send a remote lock command to the device. This locks the screen and requires the user's passcode to unlock it.

If a user has simply forgotten the passcode, an MDM server can remove it from the device and prompt the user to create a new one within 60 minutes.

Supported management commands

Managed settings

- Install Configuration Profile
- Remove Configuration Profile
- Data roaming
- Voice roaming (not available on all carriers)

Managed apps

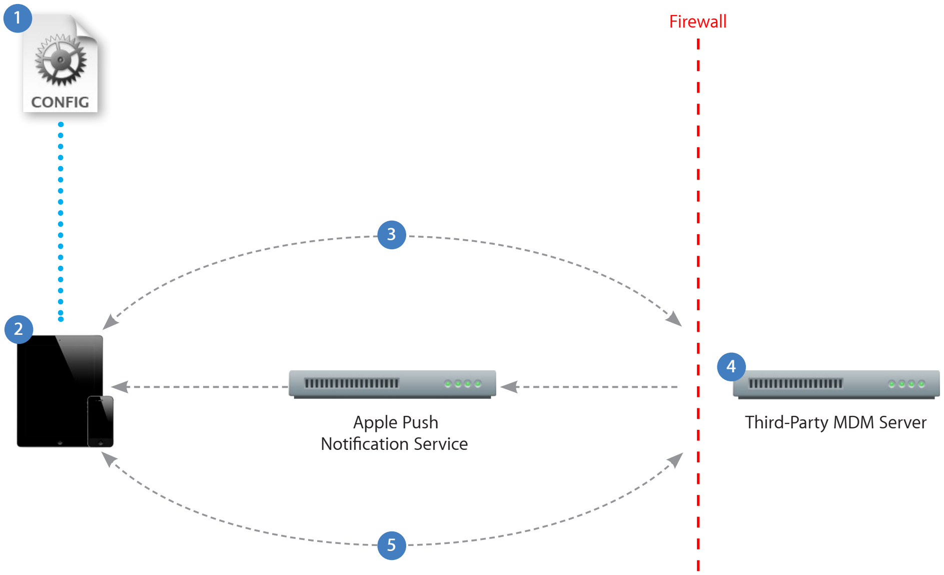
- Install managed app
- Remove managed app
- List all managed apps
- Install Provisioning Profile
- Remove Provisioning Profile

Security commands

- Remote wipe
- Remote lock
- Clear passcode

Process Overview

This example depicts a basic deployment of a Mobile Device Management server.



1 A Configuration Profile containing Mobile Device Management server information is sent to the device. The user is presented with information about what will be managed and/or queried by the server.

2 The user installs the profile to opt in to the device being managed.

3 Device enrollment takes place as the profile is installed. The server validates the device and allows access.

4 The server sends a push notification prompting the device to check in for tasks or queries.

5 The device connects directly to the server over HTTPS. The server sends commands or requests information.

For more information on Mobile Device Management, visit www.apple.com/business/mdm.

October 2011 L422501B