Strategies for iOS Deployment, Integration, and Control

Enterprise iPhone and iPad Administrator's Guide

Charles Edge

Apress®

Chapter 9

Mass-Deploying Devices

At this point, you have spent a large part of the book learning how to perform tasks on iOS-based devices, interconnect those devices to other systems, and even perform a little light software development. In Chapter 8, you created a profile by using the iPhone Configuration Utility. In this chapter, you will take those profiles and deploy them to actual mobile devices, allowing for your initial pilot and/or deployment to occur.

As shown in Chapter 8, you can deploy profiles to mobile devices in smaller environments by using the iPhone Configuration Utility. However, that strategy can be somewhat difficult for larger environments. In mass deployments, you want to keep the number of times you need to "touch" a device to a minimum. But given that each device needs to be plugged into a machine running iTunes, deploying profiles over a cabled connection represents a second "touch," and a lengthy one at that.

In this chapter, you will look at products that ease the deployment and management of iOS on a large scale. This starts with understanding the terms that will be a recurring theme throughout the chapter. We then turn our attention to some options for organizations that would like to build their own solution. But we then switch our focus to AirWatch, JAMF's Casper Suite, Dell's KACE appliance, MobileIron, and TARMAC. These products provide the most seamless and labor-free deployment that can be had, along with the best long-term management strategies.

After you've looked at each of the products, you'll then look at how to remove them. After all, I hope you will test each of them before you purchase one!

NOTE: The products in this chapter are listed alphabetically, so the author does not play favorites with any. All products have their merits and specialties, which are showcased throughout this chapter.

Deployment Terminology

iOS 4 adds Mobile Device Management (MDM) to the picture, which provides additional resources for developers to build tools that seamlessly integrate into the enterprise. However, Apple does not currently sell an end-to-end solution that can be used to deploy iOS-based devices over the air. Being able to generate management profiles dynamically and then to composite profiles based on group membership becomes, for the most part, something that most organizations will rely on a third-party product to do. Each of the products included in this chapter can leverage MDM for this task.

One of the most important aspects of MDM is that it does not require an application to be installed. When a device needs to be enrolled with the server, the server will send a message (e-mail or text, according to which product you are using) to the device. The server can also be accessed via a web portal that is supplied to users. The end user will then provide a username and password (profiles are specific to users) and choose to accept the profile or not. If accepted, the profile will be installed. You can then look at the profile and see what kind of control the profile's server can exert over your device.

NOTE: Technically, the profiles must be accepted on the device. This can happen prior to the device being supplied to an end user, or the profile can be accepted by the end user.

MDM works hand in hand with Simple Certificate Enrollment Protocol (SCEP). Apple has supported the ability to allow mobile devices to enroll into an existing SCEP server for some time. SCEP is a protocol that was developed by Cisco and is most commonly used to provide certificate services for routers, VPNs, and other devices. In Chapter 8, we discussed the importance of certificates as they relate to building and managing profiles for iOS-based devices. Therefore, it will be no surprise that Apple has chosen a protocol in such alignment with their own methodologies to automate the management of certificates.

Building Profiles from Scripts

The iPhone Configuration Utility is used to "image" iPhone, iPad, and iPod Touch. You aren't laying bits down as you would in a traditional imaging scenario. Instead, you are sending a profile and possibly some applications to the device. This is done through a configuration profile, which is a property list, prefixed with a `.mobileconfig` extension.

The iPhone Configuration Utility stores its data in the `~/Library/MobileDevice` directory. Here you will find two subdirectories:

- `Devices` contains the device data for each device that has been docked to the iPhone Configuration Utility.
- `Configuration Profiles` contains the profiles that you will assign to devices in the form of `.mobileconfig` plists.

Both of these can be managed from the command line and therefore generated en masse. First, let's look at creating devices and then configuration profiles.

Creating Devices

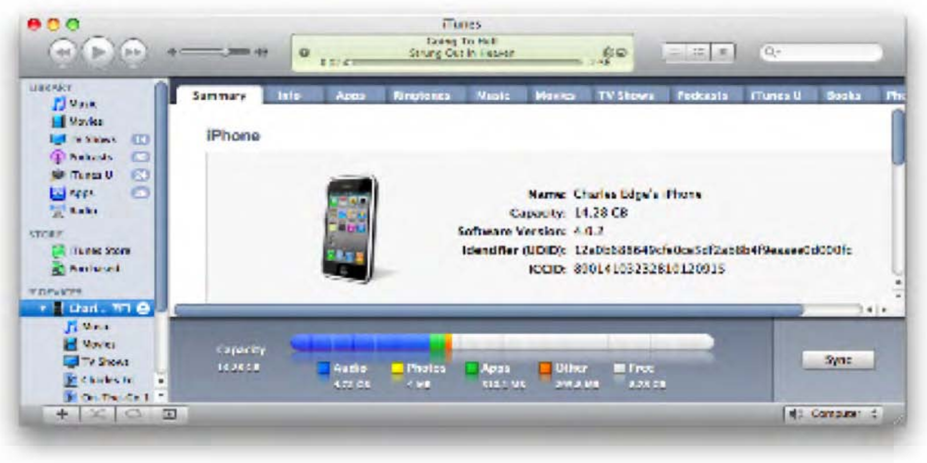
If you go into the `Devices` directory, you will see a `.deviceinfo` file for each device that you have interacted with through the iPhone Configuration Utility. Each file is prefixed by the unique device identifier (UDID) of the device. You can view each `.deviceinfo` file as a standard property list, which appears in a very simplistic fashion, as shown in Listing 9–1.

Listing 9–1. ***A Sample*** *`.deviceinfo`* ***File***

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs\
/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>UniqueChipID</key>
<integer>0</integer>
<key>applicationDictionaries</key>
<array/>
<key>configurationProfiles</key>
<array/>
<key>deviceActivationState</key>
<string>WildcardActivated</string>
<key>deviceBuildVersion</key>
<string>7E18</string>
<key>deviceCapacityKey</key>
<integer>15333203968</integer>
<key>deviceIdentifier</key>
<string>12a0b688649cfe0ce5df2ab8b4f9eaaee0d000fc</string>
<key>deviceLastConnected</key>
<date>2010-05-27T00:17:17Z</date>
<key>deviceName</key>
<string>Charles Edge's iPhone</string>
<key>devicePhoneNumber</key>
<string>1 (310) 555-1212</string>
<key>deviceProductVersion</key>
<string>3.1.3</string>
<key>deviceSerialNumber</key>
<string>12345678901</string>
<key>deviceType</key>
<string>iPhone</string>
<key>provisioningProfiles</key>
<array/>
</dict>
</plist>
```

To find a UDID, you can plug a device into iTunes, select the device from the Devices list, and then click the Summary tab. You can then click the bold Software Version to see the build version. You can click the bold Phone Number to see the international mobile equipment identity (IMEI), or double-click to see the integrated circuit card identifier (ICCID) of the SIM card. Click the bold Serial Number to see the identifier (or UDID). This is shown in Figure 9–1.

> **NOTE:** ICCID, IMEI, and UDID are unique to the device, and each has a different purpose. The ICCID is attached to the SIM card of a device. The IMEI is for GSM (Global System for Mobile Communications) connectivity. The UDID is a unique number used exclusively for iOS-based devices, most commonly used for provisioning software.



Name: Charles Edge's iPhone
Capacity: 14.28 GB
Software Version: 4.0.2
Identifier (UDID): 12a0b5d8649cfe0ca5cf2ab8b4f9eaaaeCdC00fc
ICCID: 89014103232810120015

Figure 9–1. *Obtaining the UDID of a device*

The serial number can also be obtained from a bar code on the box that came with the device, although the UDID cannot at this time. The serial number, though, can then be brought into a database that has both, to correlate them (assuming you are programmatically going to wrangle this data at a later time) and assign profiles based on, for example, Open Directory or Active Directory group membership of the primary user.

You can copy a template file without unique identifiers and then use defaults to put the unique data into the file. Or you can use PlistBuddy to create a file from scratch. The data can then be viewed in somewhat of a 2D fashion up to this point, as shown with a sample CSV file (comma separated value file) in Figure 9–2. The problem then comes in the arrays, because in addition to referencing data from the `.mobileconfig` files we'll look at in a moment, these arrays are using localized plists to form a relational context to data.

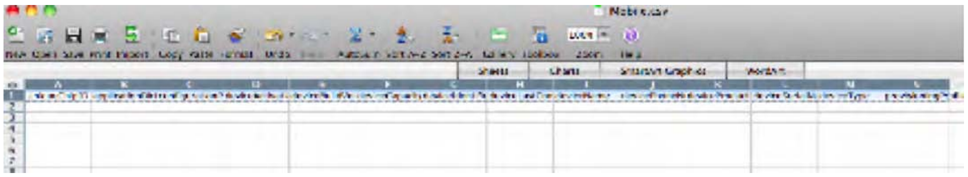


Figure 9–2. *Required fields*

Creating Configuration Profiles

You can then look at a `.mobileconfig` file, which appears in a very simplistic form, as shown in Listing 9.2.

Listing 9–2. *A Sample `.mobileconfig` Profile*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs\
/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>PayloadContent</key>
 <array>
 <dict>
 <key>FullScreen</key>
 <false/>
 <key>IsRemovable</key>
 <true/>
 <key>Label</key>
 <string></string>
 <key>PayloadDescription</key>
 <string>Configures Web Clip</string>
 <key>PayloadDisplayName</key>
 <string>Web Clip</string>
 <key>PayloadIdentifier</key>
 <string></string>
 <key>PayloadOrganization</key>
 <string></string>
 <key>PayloadType</key>
 <string>com.apple.webClip.managed</string>
 <key>PayloadUUID</key>
 <string>80222944-B43C-4A43-AB93-2998CDCBE808</string>
 <key>PayloadVersion</key>
 <integer>1</integer>
 <key>Precomposed</key>
 <false/>
 <key>URL</key>
 <string></string>
 </dict>
 </array>
 <key>PayloadDescription</key>
 <string>Profile description.</string>
 <key>PayloadDisplayName</key>
 <string>Profile Name</string>
 <key>PayloadOrganization</key>
 <string></string>
 <key>PayloadRemovalDisallowed</key>
 <false/>
 <key>PayloadType</key>
 <string>Configuration</string>
 <key>PayloadUUID</key>
 <string>5B0879F3-9BA9-41E7-AC8F-F4703D4400DB</string>
 <key>PayloadVersion</key>
 <integer>1</integer>
</dict>
</plist>
```

You can then create a single `.mobileconfig` file, make it a template, and match the settings for your template user, per group. Those `.mobileconfig` files then get applied to each device, which you could do in batches. You can also dynamically copy the files to a web server and send an SMS (Short Message Service)/e-mail to the user, who could then click on the files to apply them or to dock the device.

You can also add applications by using a preexisting array and copying it, although if there are licensing concerns regarding the application you are distributing, it would be wise to investigate the ramifications of doing so first. Keys and such are defined in the *iPhone OS Enterprise Deployment Guide* (Apple, 2010) along with sample AppleScript for creation of files.

You can also copy the database by copying the property list files between machines. When the iPhone Configuration Utility is opened, it will automatically read in the new property lists and display the information. Overall, this process is going to be as much work as dynamically generating provisioning on the fly via the Ruby sample code provided by Apple, while netting you less of a result (unless of course, your shell scripting powers are demigod-like).

Apple's Sample Code

Although Apple doesn't provide an end-to-end solution, the makers of iOS do provide a fair amount of sample code for developers. This means Apple gives you the APIs, but that you will need developers to complete the solution.

It seems as though each year at Apple's Worldwide Developers Conference (WWDC) there is a lot of talk about new sample code, or about APIs released by Apple. Among these is the Ruby sample code referenced briefly in Chapter 1. Using this code as a starting point, an in-house development staff can build a tool that can be used to enroll mobile devices into your environment, creating a configuration profile as you go. Again, to leverage the sample code, you will need to develop the tool.

The sample code uses Ruby for auto-enrollment. This code is available at `http://developer.apple.com/library/ios/#documentation/NetworkingInternet/Conceptual/iPhoneOTAConfiguration/Introduction/Introduction.html`. Apple documents the mail, network, and policy settings here, and then goes on to cover certificate management, including certificate revocation information.

MDM manages iOS-based devices after they have initially been deployed. Similar to policy management, the goal of most MDM environments is to apply configuration files that in turn apply policies. This means that each tool will need an agent that can then communicate back to a centralized server and exchange configuration profiles as a part of the client-server relationship.

As with most other things in the IT industry, you can do practically anything you want, including developing your own MDM-based enrollment solution. However, many organizations have specific strategies indicating when to build a custom solution from the ground up and when to buy software that is officially supported by a vendor. All strategies have merits. We will not go into each here, but whatever strategy your organization employs for other solutions is applicable here as well.

For more information on the structure, strategy, and other big-picture concepts, regard the sample code provided by Apple at `http://images.apple.com/iphone/business/docs/iPhone_OTA_Enrollment_Configuration.pdf`.

Given that this book is not a title on Ruby or any other programming language, at this point we will look at third-party packages that can aid you in your mass deployment.

AirWatch

AirWatch is a tool that can be used to deploy profiles over the air, en masse. AirWatch runs on Windows servers in a .NET environment, currently leveraging Silverlight to provide a stellar experience with the management console and using the latest in Apple's MDM APIs to provide an even more stellar result to end users. AirWatch is mature, robust, and comes with several unique options, such as the capability to import devices from a CSV file and then automate sending SMS alerts to devices to enroll.

AirWatch can be acquired either as a software as a service (SaaS) solution or with an on-premises server. Questions about how to license or host a product are typically best left to the vendor, but suffice it to say that the choice between the two is a typical quandary for many environments and many types of technologies, ranging from messaging to files. Complexity of network environments, decision on pricing models, exposure of data to an outside vendor, and other factors can help play a role in the decision-making process.

Whether you choose to host the server at your location or outsource it to AirWatch, the configuration is likely not a "customer installable" process. This means that you will more than likely engage AirWatch to set up the solution for you (which is pretty similar to every other product mentioned throughout this chapter). Therefore, this section focuses on the use and management of the devices through the web portal after the solution is configured.

Managing Objects in the Portal

The first time you log into the AirWatch portal as an administrator, you will want to perform several tasks. These include creating users, creating location groups, creating a default device profile (or one per location group in some cases), connecting AirWatch to some of your back-end systems, and of course when you're finally ready, enrolling a device to be managed by the AirWatch agent.

Setting Up Administrative Users

We will get started with the global settings. If you require additional administrative accounts, you should create them. You can do so by using the Configure option in the AirWatch portal. Hovering over Configure brings up the User Setup option (Figure 9–3).

Clicking the User Setup option then brings up a screen enabling you to Search Users, check Login History, define Roles, and Add Users.

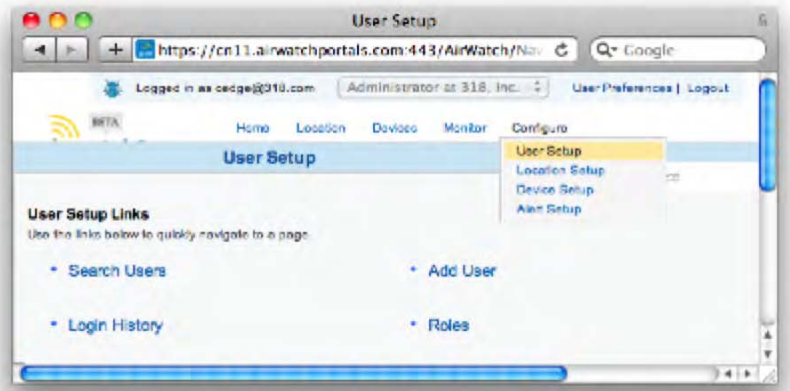
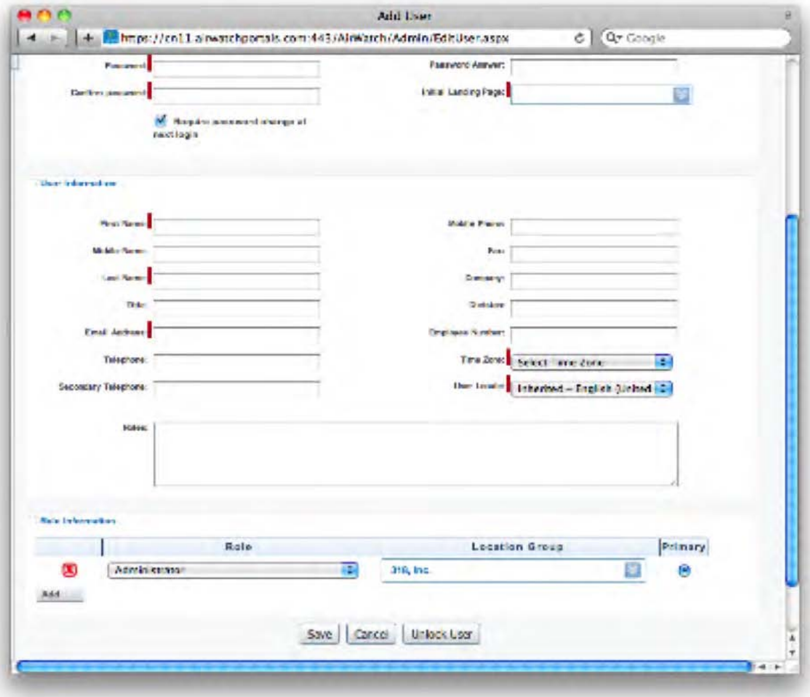

User Setup
https://cn11.airwatchportals.com:443/AirWatch/Nav
Google
Logged in as cedge@318.com
Administrator at 318, Inc.
User Preferences | Logout
BETA
Home
Location
Devices
Monitor
Configure
User Setup
Location Setup
Device Setup
Alert Setup
User Setup Links
Use the links below to quickly navigate to a page
Search Users
Add User
Login History
Roles

Figure 9–3. *Setting up administrators in AirWatch*

You will then see a screen asking for information about the user (Figure 9–4). Pay particular attention to the Role field, which sets the user as either an administrator or a reviewer over a given location group (and its children). Setting a user as an administrator provides that user full control over the location group that the setting was created for. After you have filled out the fields to generate the new user, click the Save button.

As with most other aspects of technology, it is recommended that administrative users have permission to access only items they need. Creating location groups and delegating permissions through them will enable you to maintain granular controls over your delegated administrative access much in the same way that an organization unit will enable you to do so in Active Directory.



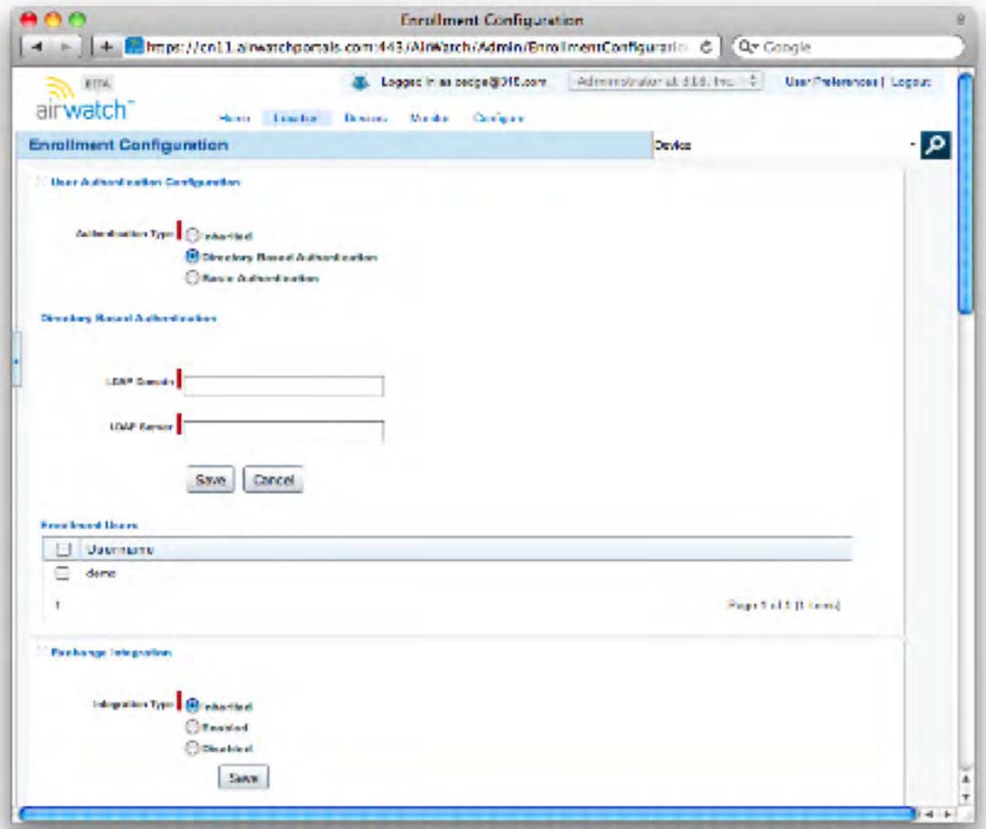
Time Zone: Select Time Zone
User Locale: Inherited – English (United
Role: Administrator
Location Group: 3rd, Inc

Figure 9–4. *User options and roles in AirWatch*

Configuring LDAP and SCEP

Next, you will want to configure what AirWatch refers to as *location groups*. Each location group can have its own Lightweight Directory Access Protocol (LDAP) and SCEP implementation. If you have an SCEP server (introduced earlier in this chapter), you will want to configure AirWatch to communicate with your server for certificate management. The SCEP server would then need to be accessible from the Internet provided that you will have devices managed when they are not on your network. LDAP is used to perform enrollment authentication requests; therefore, the LDAP server for your environment does not necessarily need to be exposed in the same fashion.

Configuring LDAP and SCEP are done on the same screen. Click Location in the top menu bar and then click the option for Location Groups. You will then see a list of all of the location groups that the user you are authenticated as can administer. Click the location group to manage and then click the Enrollment Configuration button. Next you will see the User Authentication configuration screen, where you can set up an LDAP connection, as you can see in Figure 9–5.


Enrollment Configuration

User Authentication Configuration

Authentication Type: Inherited [✓ Directory Based Authentication] Basic Authentication

Directory Based Authentication

LDAP Domain:

LDAP Server:

Save Cancel

Enrolled Users

	Username
☐	demo

Exchange Integration

Integration Type: [✓ Inherited] Enabled Disabled

Save

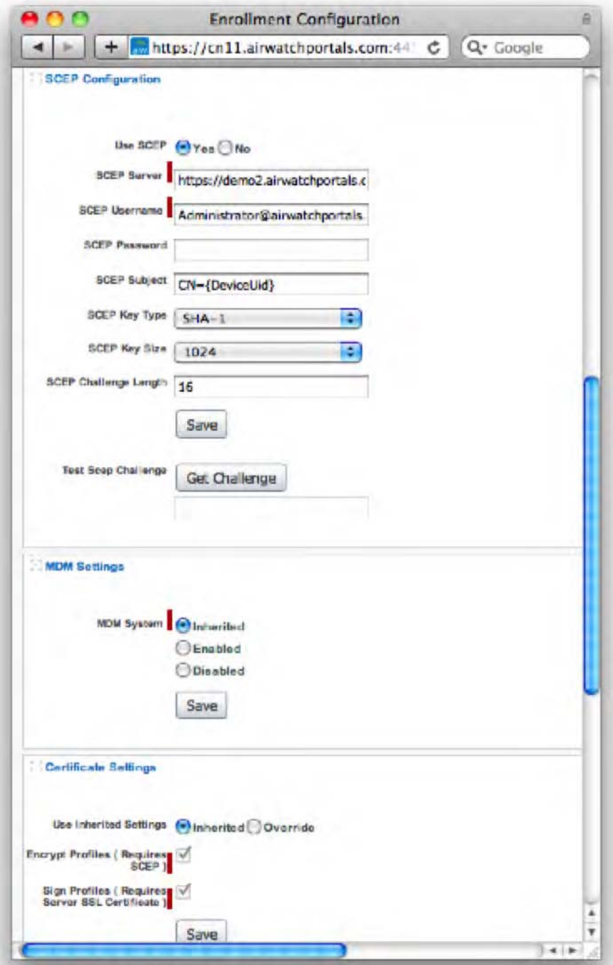
Figure 9–5. *Linking AirWatch to your LDAP environment*

Selecting the Directory Based Authentication option will allow you to enter an LDAP Domain and an LDAP Server. These will need to be supplied by the network administrator for the LDAP environment. The LDAP Server can be a name or an IP address, with a prefix (URI specification) of `LDAP`. For example, if you have a domain controller or an Open Directory server sitting at `192.168.55.2`, you could use `LDAP://192.168.55.2`. The domain can be used in Active Directory environments to signify the Active Directory domain name.

NOTE: If you have an Exchange environment, you will want to enable Exchange Integration for AirWatch at this screen as well (which is not the default setting).

Next, scroll down to the SCEP Configuration section of the page. You will see the options that should correspond to your SCEP implementation (Figure 9–6). Match Key

Sizes, Key Types, SCEP Subject, SCEP Password, and SCEP Username with the address that you fill into the SCEP Server portion of the screen. For most environments, the remainder of the settings on this page will suffice.



SCEP Configuration

Use SCEP	[✓ Yes] No
SCEP Server	https://demo2.airwatchportals.c
SCEP Username	Administrator@airwatchportals
SCEP Password	
SCEP Subject	CN={DeviceUid}
SCEP Key Type	SHA-1
SCEP Key Size	1024
SCEP Challenge Length	16
	Save
Test Scep Challenge	Get Challenge

MDM Settings

MDM System	[✓ Inherited] Enabled Disabled
	Save

Certificate Settings

Use Inherited Settings	[✓ Inherited] Override
Encrypt Profiles (Requires SCEP)	✓
Sign Profiles (Requires Server SSL Certificate)	✓
	Save

Figure 9–6. *Configuring SCEP settings in AirWatch*

After you click Save, we will move on to creating the profiles that can then be pushed out to clients.

Creating a Profile

Right now you are probably thinking that you were told that all of the 3rd party products for iOS mass deployment use the iPhone Configuration Utility. Well, close. AirWatch has done something pretty intelligent with their configuration profiles: duplicating what Apple did, but giving more options customized for AirWatch (and more important, automatic enrollment and configuration of applications).

Each location group should have a profile (or they can all inherit the profile from the parent location group). To create the profiles, click the Devices option and then click Device Profile Management, which brings up a screen similar to that in Figure 9–7.

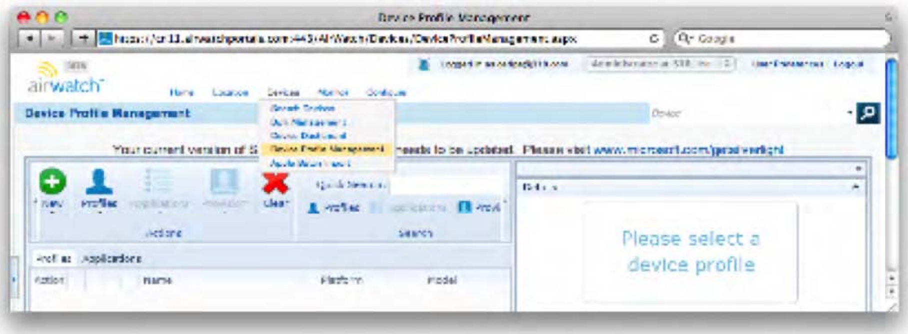

Device Profile Management
airwatch
Device Profile Management
Device Profile Management
New
Profiles
Clear
Actions
Search
Profiles
Applications
Action
Name
Platform
Model
Please select a device profile

Figure 9–7. *Managing profiles in AirWatch*

Click the New button and then select Managed Profile from the options, which brings up a screen with an option to set the platform (Figure 9–8).

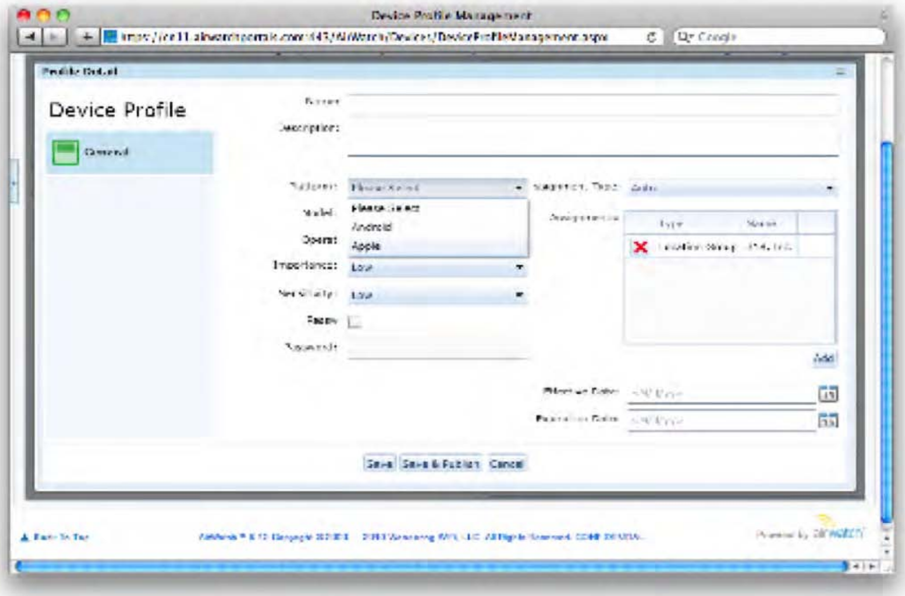
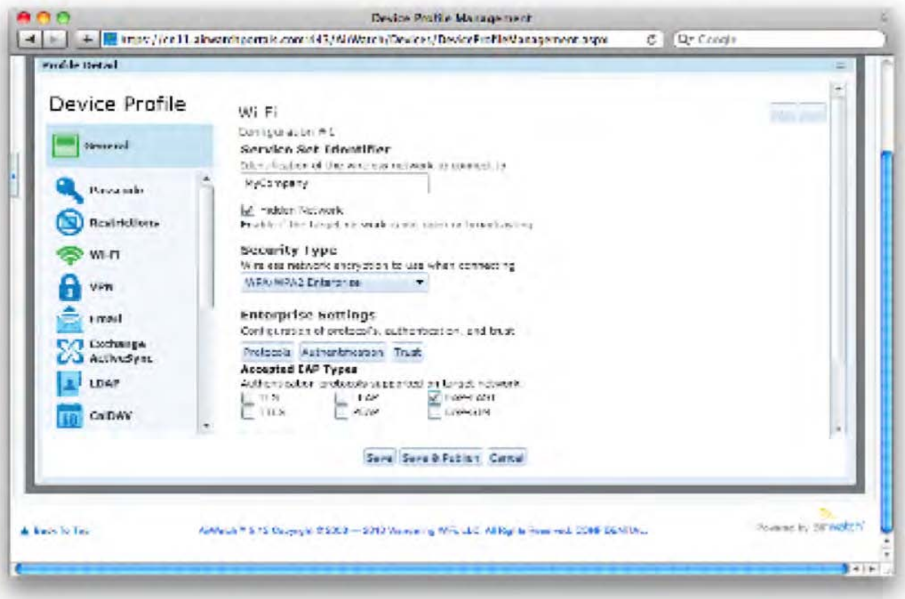


Figure 9–8. *Setting the Platform field for AirWatch device profiles*

Set the Platform field to Apple and then you will see a screen that will look familiar, as it is based on the options available in the iPhone Configuration Utility. Configure the options as you would like (based on the walk-throughs available in Chapter 8) and then click Save when you are finished (Figure 9–9).



Wi Fi

Configuration #1

Service Set Identifier

Identification of the wireless network to connect to

MyCompany

☑ Hidden Network

Enable if the target network is not open or broadcasting

Security Type

Wireless network encryption to use when connecting

WPA/WPA2 Enterprise

Enterprise Settings

Configuration of protocols, authentication, and trust

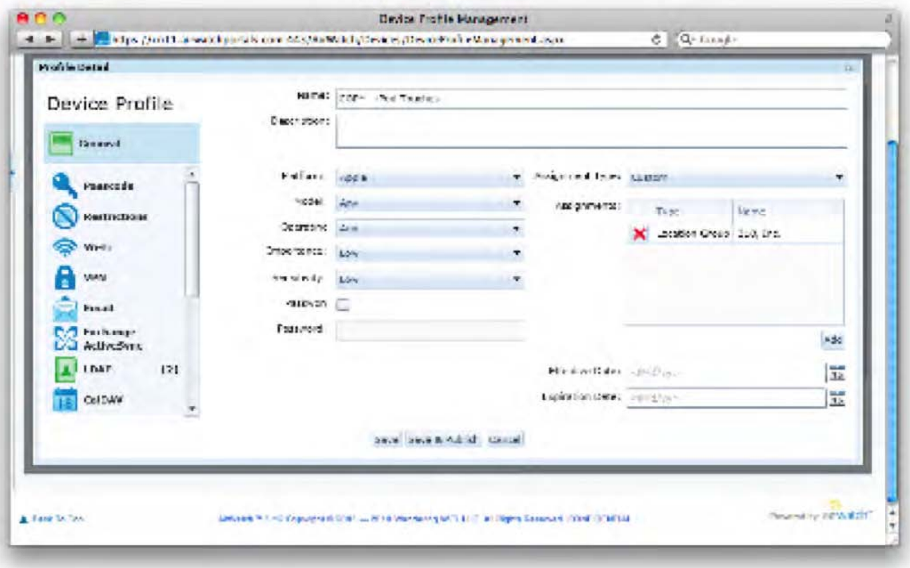
Accepted EAP Types

Authentication protocols supported on target network

☐ TLS ☐ LEAP ☑ EAP-FAST
☐ TTLS ☐ PEAP ☐ EAP-SIM

Figure 9–9. *iPhone Configuration Utility-like profile options*

Profiles can be assigned to a location group, and systems newly enrolled into that location group will take on the profile (Figure 9–10).



Name: [illegible]
Description:
Platform: Apple
Model: Any
Ownership: Any
Importance: Low
Sensitivity: Low
Password:
Assignment Type: Custom
Assignments:

Type	Name
Location Group	[illegible]

Effective Date:
Expiration Date:

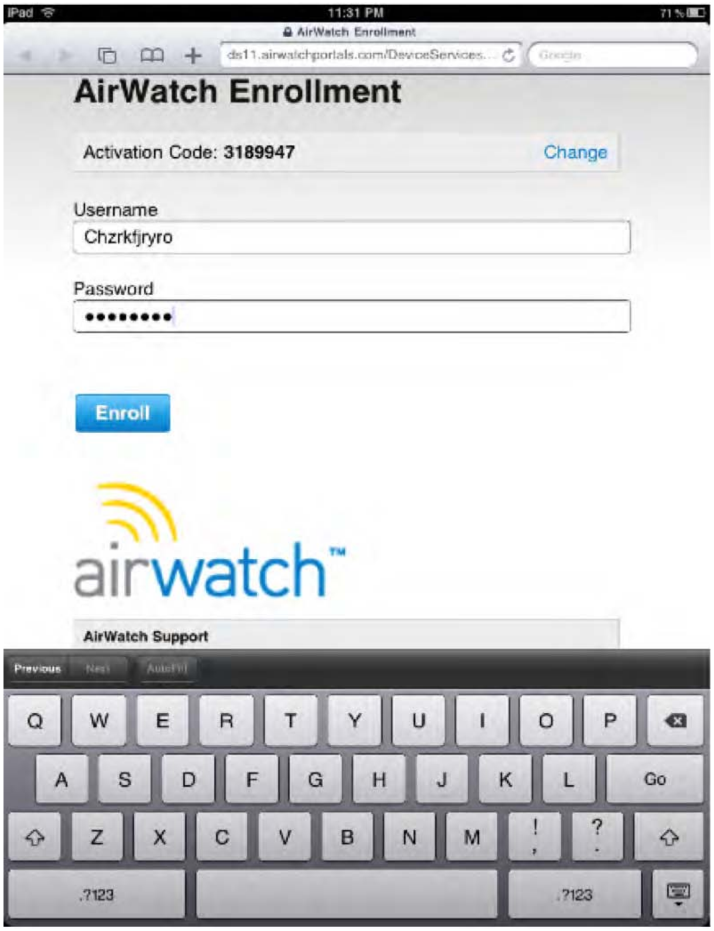
Figure 9–10. *Configuring the devices a profile will be assigned to*

> **NOTE:** You can also create a location within a location group and assign a profile to just the location.

In order to see the assignment of these profiles via MDM in action, let's enroll the first client system to the server. You are now ready to enroll a device so it can be associated with the appropriate profile.

Enrolling a Device

The auto-enrollment process for AirWatch is straightforward to end users. They are sent a link (typically through SMS) and they then click that link. At the login prompt, the user will log in as himself or herself, thus allowing the profile to be personalized to that user account.



AirWatch Enrollment

Activation Code: **3189947** Change

Username: Chzrkfjryro

Password: •••••••••

Figure 9–11. *Enrolling for Airwatch-based MDM*

After the user is enrolled, you will then see the device listed in the portal (Figure 9–12).

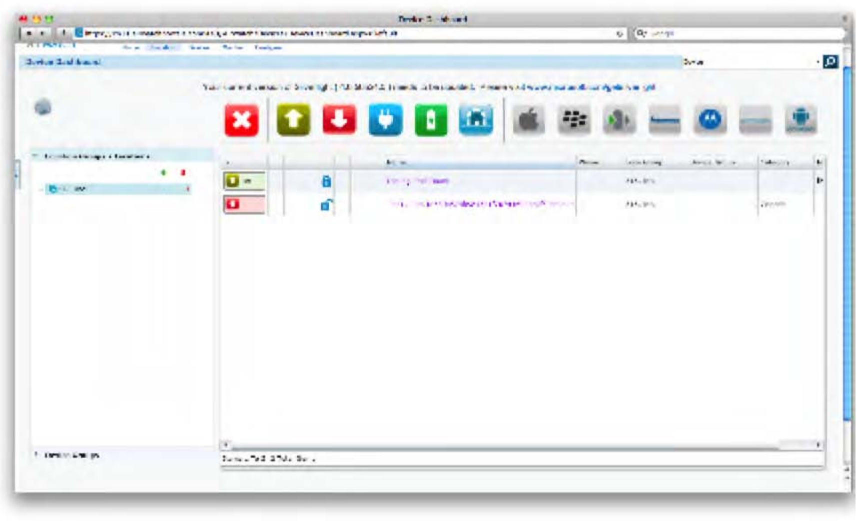


Figure 9–12. *Showing enrolled devices*

During the enrollment, you will be prompted to install the configuration profile. The user of the device must choose to accept the installation, or the process will end at this point. As a part of the enrollment process, AirWatch will then be able to manage the device remotely via MDM. You can use the icons along the top of the screen (Figure 9–13) to manage the device.

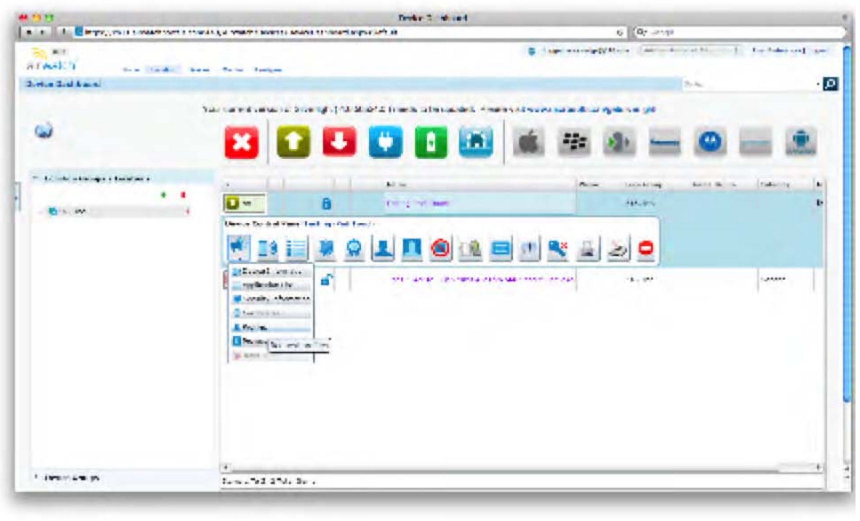


Figure 9–13. *Device management options*

Here you also find one of the most popular options: to remotely wipe a device. You have a few options for doing so, including using the web portal for Microsoft Exchange, as was shown in Chapter 4. The ability to do so through AirWatch (Figure 9–14), as well as unlock the passphrase to wake up a sleeping phone and lock the device, is a welcome addition to many environments.

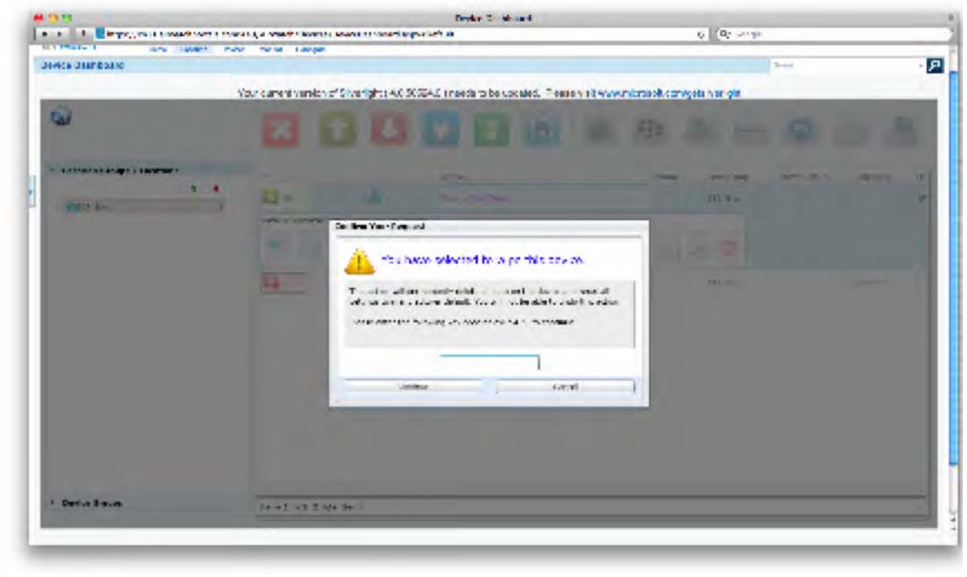


Figure 9–14. *Wiping devices by using AirWatch*

JAMF's Casper Suite

JAMF Software has a number of tools for imaging the Mac OS X platform. Among these are the Recon Suite, for obtaining an inventory of hardware and software assets, and the Casper Suite, a full-featured client management solution for Mac OS X that has been extended to include the management of iOS-based devices. The Casper Suite uses a server, referred to as the JAMF Software Server (JSS), for managing the database of devices, packages, automations, provisioning profiles, applications, and other information that is tracked and leveraged for management and imaging. JAMF also has a tool called the Imaging Suite, which is used for imaging computers, and another called Composer, used to build packages for deployment to the Mac OS X platform. However, neither of these is required for iOS-based devices and so they are not further covered in this chapter.

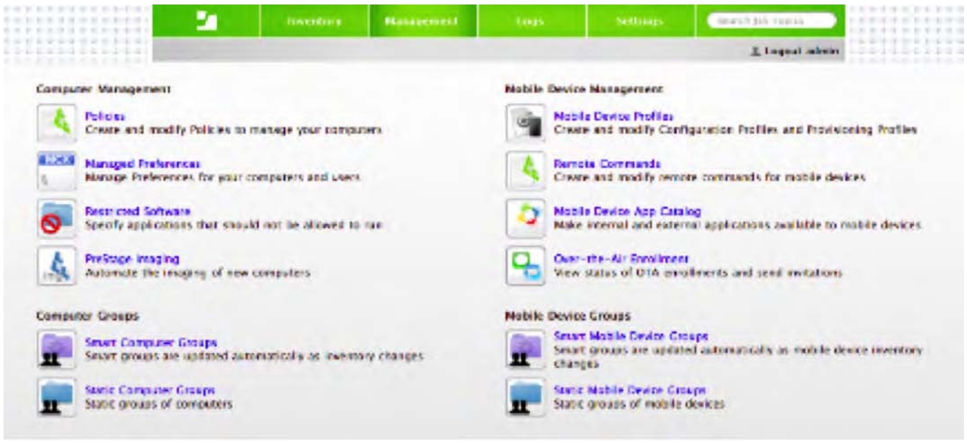
If you are already using the Casper Suite, the learning curve to start deploying iOS-based devices into your environment will be negligible. Provided you have a Casper Suite installation for a pilot, or have purchased the software and are currently using it on the Mac OS X platform, you can extend the use of your Casper Suite deployment. If you do not have a JSS for a pilot, you can use the MDM-based services on a cloud-based pilot provided from JAMF.

The main tasks to prepare your environment for mass management and deployment of iOS devices if you will be leveraging Casper for enrolling, configuring, managing, and tracking devices includes the following:

- Setting up the JSS to interact with SCEP, LDAP, and other aspects of your environment
- Logically grouping devices
- Creating configuration profiles to associate with devices or groups of devices
- Enrolling devices into Casper
- Adding applications to the catalog
- Managing devices

NOTE: We will look at these main tasks, but as with AirWatch in the previous section of this chapter, the Casper Suite can do much more than the common tasks covered in this section.

If you log into your JAMF Software Server and click the Management tab, you will see a screen similar to that in Figure 9–15.



Computer Management

Policies
Create and modify Policies to manage your computers

Managed Preferences
Manage Preferences for your computers and users

Restricted Software
Specify applications that should not be allowed to run

PreStage Imaging
Automate the imaging of new computers

Computer Groups

Smart Computer Groups
Smart groups are updated automatically as inventory changes

Static Computer Groups
Static groups of computers

Mobile Device Management

Mobile Device Profiles
Create and modify Configuration Profiles and Provisioning Profiles

Remote Commands
Create and modify remote commands for mobile devices

Mobile Device App Catalog
Make internal and external applications available to mobile devices

Over-the-Air Enrollment
View status of OTA enrollments and send invitations

Mobile Device Groups

Smart Mobile Device Groups
Smart groups are updated automatically as mobile device inventory changes

Static Mobile Device Groups
Static groups of mobile devices

Figure 9–15. *Using the JSS*

NOTE: Throughout the remainder of this section on the Casper Suite, I will simply refer to this location as *the Management tab*, given that it will be the starting navigation point for several tasks.

The options available for iOS-based systems (deployment and MDM) are listed in the right column, whereas the options available for the Mac OS X platform are listed on the left side of the screen. The Mobile Device Management options include the following:

Mobile Device Profiles: Enables you to create and edit configuration and provisioning profiles

Remote Commands: Enables you to lock screens, remotely wipe devices, and reset passcodes for iOS-based devices

Mobile Device App Catalog: Enables you to define internal applications and applications from the App Store that are then deployed over the air

Over-the-Air Enrollment: Sends invitations for computers to enroll with the server via SMS or e-mail

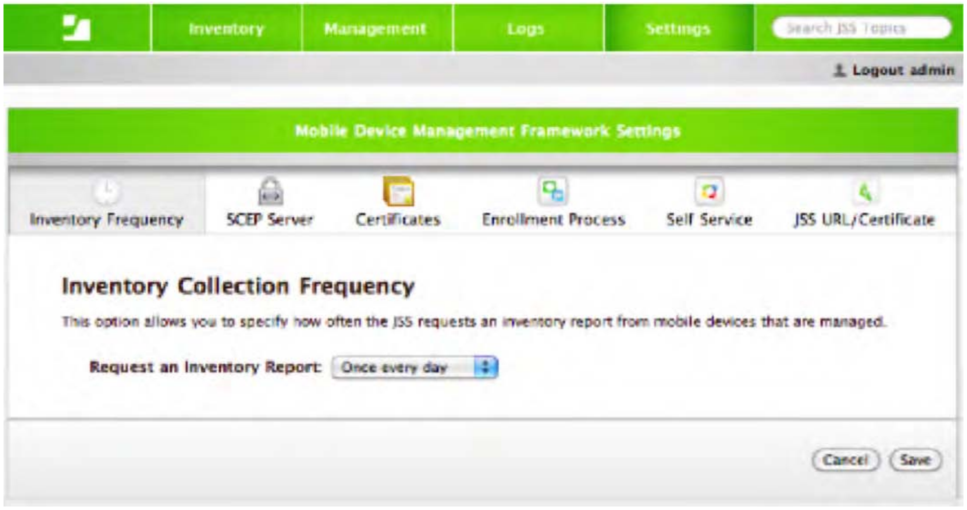
Smart Mobile Device Groups: Creates dynamic groups based on the contents of fields used in the JSS (for example, version of iOS, user, groups, locations, and so forth)

Static Mobile Device Groups: Creates groups by adding devices into the groups manually

Configuring Global Settings

It is usually best to start with configuring the global settings for the server itself so that you can perform meaningful tests on the devices that you are deploying. This is going to include configuring the JSS to communicate or leverage other services in your network environment. To get started, click the Settings tab at the top of the screen. Then click Mobile Device Management Framework Settings, which provides access to global settings specific to iOS-based device management. You will see options to configure the interval with which devices report to the JSS, the SCEP server, the SSL certificates that are used, how enrollment is handled, global options for self-service, and the URL to be used by devices to access the server.

The Inventory Collection Frequency defines how often the mobile device checks back in with the server. During the check-ins, the device will look to see whether new management profiles need to be applied and whether any settings are out of sync with those on the server. By default, devices will check back in with the JSS every day. To alter this, use the drop-down list in the Request an Inventory Report field (Figure 9–16) to provide a new frequency. At the next interval that the iOS-based devices have cached to check into the JSS, they will then all obtain the new interval and start checking in based on the defined time.



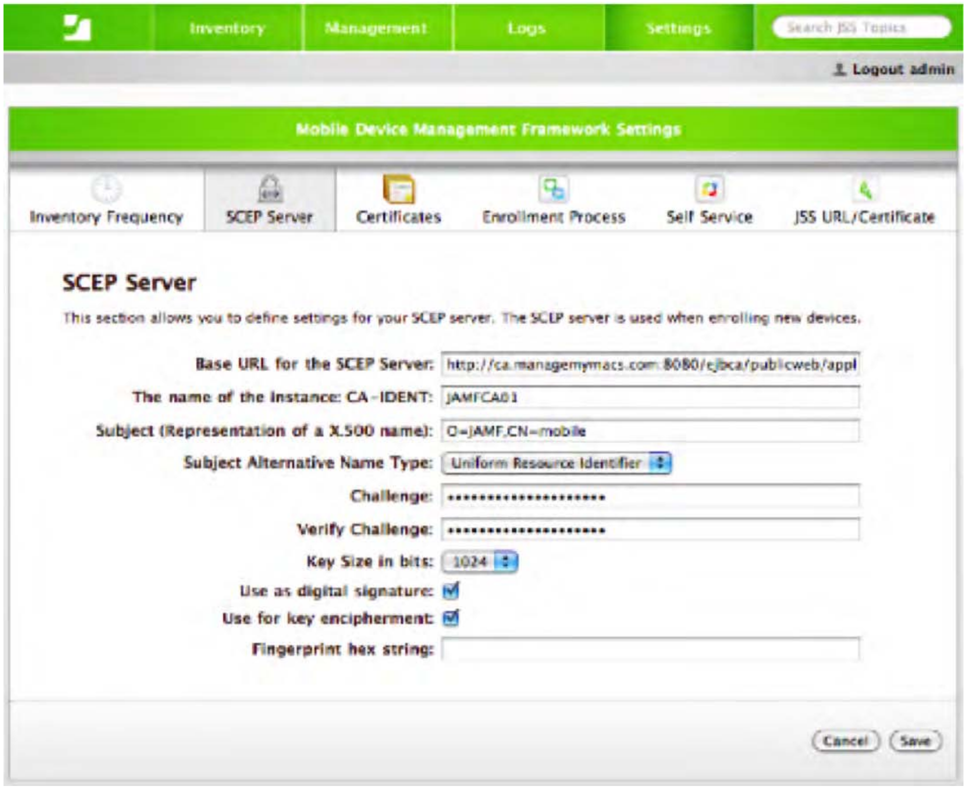
Inventory Collection Frequency

This option allows you to specify how often the JSS requests an inventory report from mobile devices that are managed.

Request an Inventory Report: Once every day

Figure 9–16. *Defining your inventory collection frequency*

As with the other products, you are going to need an instance of an SCEP server. The JSS can look to (and tell mobile devices to look to) any acceptable SCEP server that you have available in your environment. Using the options seen in Figure 9–17, you will define the URL of the SCEP server, the CA, the Subject, the Subject Type, the Challenge, the Key Size, and the Fingerprint string. You will also indicate whether the SCEP-provided certificates are used in digital signatures and whether to encrypt mobile devices. Each of these settings is provided in their corresponding fields and each is going to need to match the values available on your SCEP server.



SCEP Server

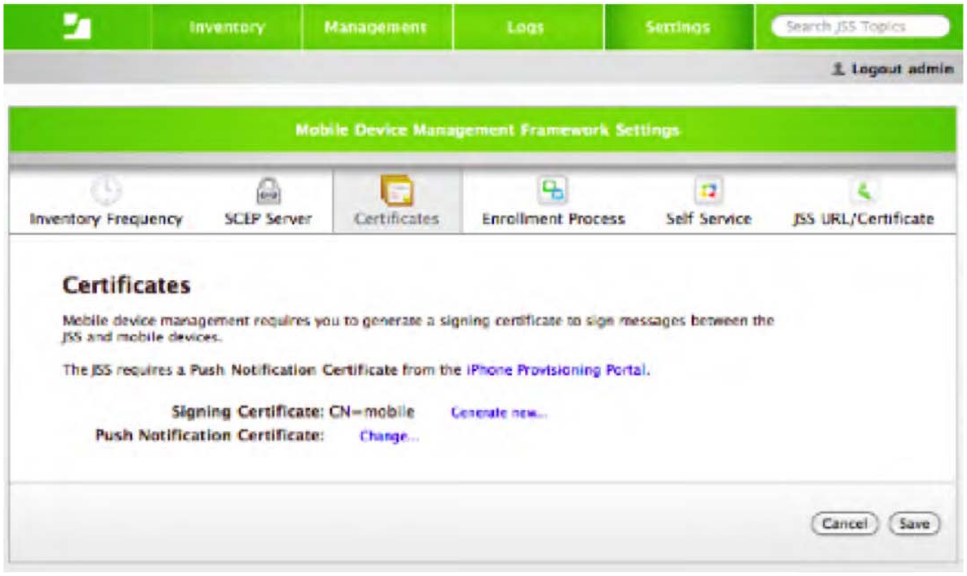
This section allows you to define settings for your SCEP server. The SCEP server is used when enrolling new devices.

Base URL for the SCEP Server: http://ca.managemymacs.com:8080/ejbca/publicweb/appl
The name of the Instance: CA-IDENT: JAMFCA01
Subject (Representation of a X.500 name): O=JAMF,CN=mobile
Subject Alternative Name Type: Uniform Resource Identifier
Challenge: ••••••••••••••••••••
Verify Challenge: ••••••••••••••••••••
Key Size in bits: 1024
Use as digital signature: ✓
Use for key encipherment: ✓
Fingerprint hex string:

Figure 9–17. *Configuring SCEP on the JSS*

Next, click Certificates and then configure a signing certificate and a Push Notification certificate. You can use an intermediary CA or most other forms of certificate signing. Click the Generate New link (Figure 9–18) to create a new signing certificate and/or the Change button to select a new certificate for Push Notification.

NOTE: Much of the MDM framework from Apple for iOS-based devices is heavily dependent upon certificates, so pay extra attention to the configuration of these.



Certificates

Mobile device management requires you to generate a signing certificate to sign messages between the JSS and mobile devices.

The JSS requires a Push Notification Certificate from the iPhone Provisioning Portal.

Signing Certificate: CN=mobile Generate new...
Push Notification Certificate: Change...

Figure 9–18. *Installing certificates for the JSS*

After you have configured your certificates, click Enrollment Process to configure the settings that apply to the end-user experience of enrolling devices into the server environment. Here, you will first decide whether devices (or accounts on devices) will need to have an invitation to join. If you have a large deployment of currently unmanaged devices and do not have the settings (UDID, and so forth) for them documented, this option is probably going to save you a lot of time while you are remediating the deployment into a managed environment. However, if you are starting off by pushing profiles to all new devices, you likely will not want to use this option. Strategically, you can use the option and then disable it after you have gotten all of the unknown devices input into the server.

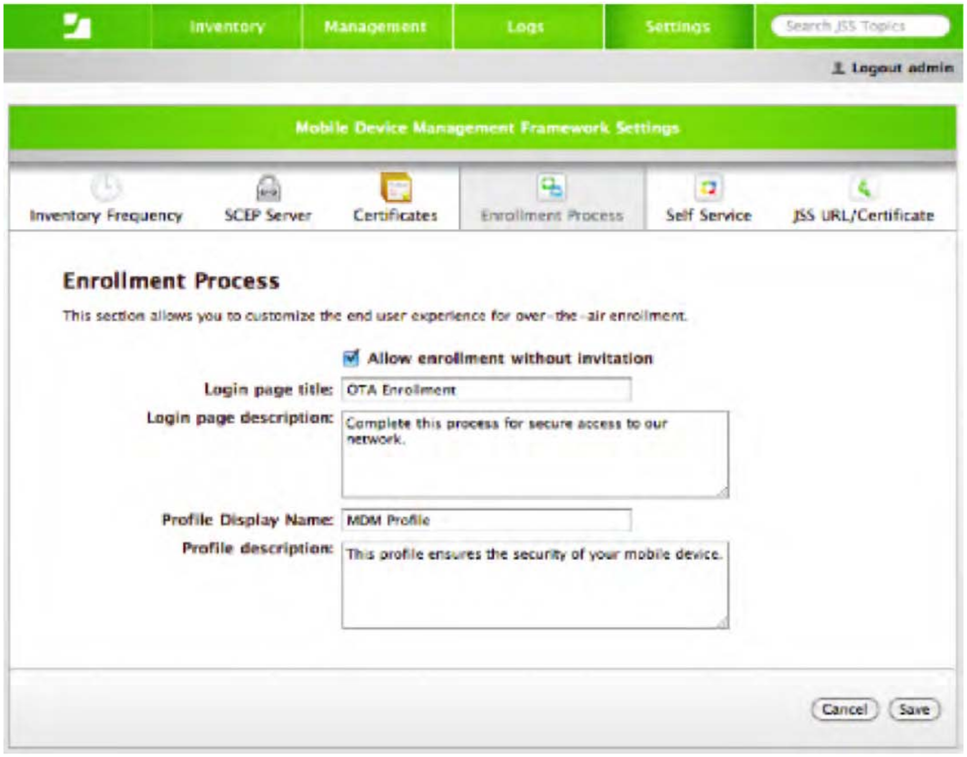
The following fields, shown in Figure 9–19, control the look and feel of the enrollment page that the end user is shown:

Login Page Title: Defaults to OTA Enrollment (Over The Air), but you can replace that with the name of your organization.

Login Page Description: Text to instruct the user as to what to do at this page.

Profile Display Name: Text to indicate what the profile is used for (or in many cases from where it came).

Profile Description: Text to display more details, shown in conjunction with the profile description (more on where these appear to users in the upcoming “Enrolling Devices” section).



Enrollment Process

This section allows you to customize the end user experience for over-the-air enrollment.

☑ Allow enrollment without invitation

Login page title: OTA Enrollment

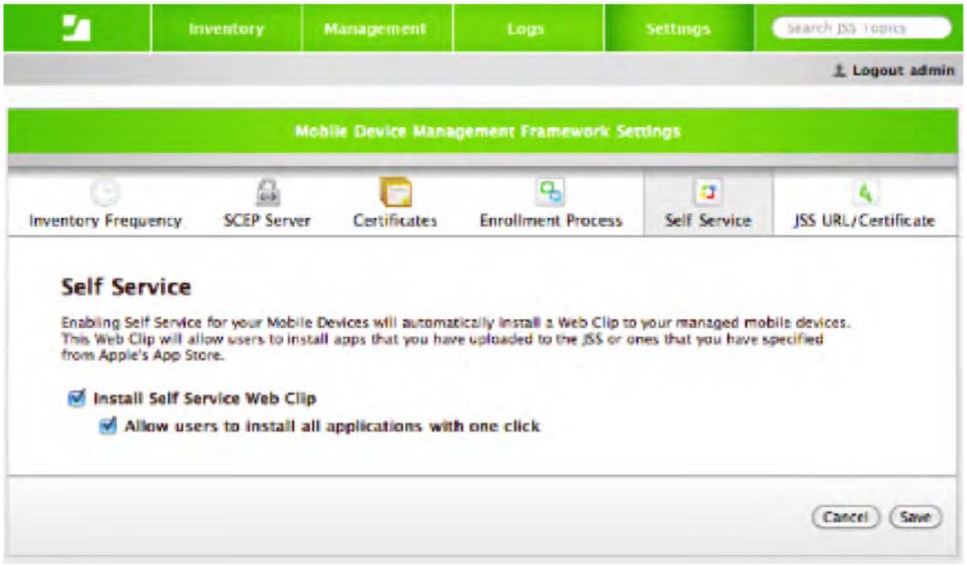
Login page description: Complete this process for secure access to our network.

Profile Display Name: MDM Profile

Profile description: This profile ensures the security of your mobile device.

Figure 9–19. *Customizing the enrollment screen and process*

After you have customized how users will see this information, click the Self Service button. Here, as shown in Figure 9–20, you can control whether the self-service portal will be shown on each user's device and whether the Install All button (shown later in Figure 9–41) will be shown to users, thus allowing them to install all recommended software with one click, rather than installing each application separately (definitely recommended if you are pushing out a lot of applications).



Self Service

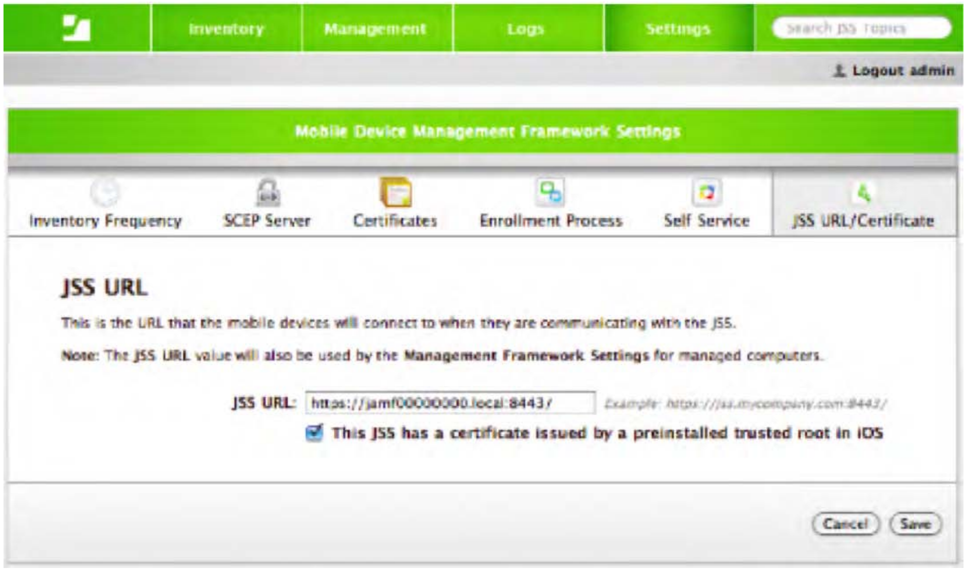
Enabling Self Service for your Mobile Devices will automatically install a Web Clip to your managed mobile devices. This Web Clip will allow users to install apps that you have uploaded to the JSS or ones that you have specified from Apple's App Store.

☑ Install Self Service Web Clip

☑ Allow users to install all applications with one click

Figure 9–20. *Choosing whether to use self-service*

Finally, click JSS URL/Certificate. Here you specify the URL that the device will use to check back into the JSS (the interval for checking in having been defined previously in the Inventory Frequency section of the Mobile Device Management Framework Settings). You can specify any URL you wish (as you can see in Figure 9–21, the default is the local host name of the system running the JSS), provided that the port and the host name actually point to the JSS. You can also indicate whether the JSS server has been outfitted with a certificate trusted by one of the root certificate authorities included with all new iOS-based devices. This allows you to verify that the communications for JSS-specific traffic are secured while in transit between the JSS and the iOS-based device.



Mobile Device Management Framework Settings

JSS URL

This is the URL that the mobile devices will connect to when they are communicating with the JSS.

Note: The JSS URL value will also be used by the **Management Framework Settings** for managed computers.

JSS URL: https://jamf00000000.local:8443/ *Example: https://jss.mycompany.com:8443/*

✓ This JSS has a certificate issued by a preinstalled trusted root in iOS

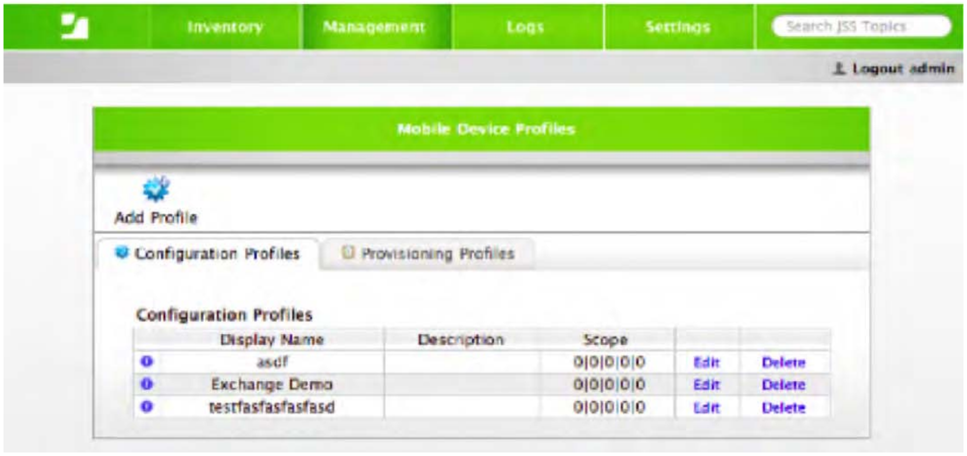
Figure 9–21. *Providing a URL for devices to access the JSS*

Click Save to commit your changes (for each screen) to the Mobile Device Management Framework. After you have configured the global settings for the JSS, you can then move on to starting to configure the more granular options, such as creating smart groups, building profiles, and ultimately enrolling devices.

Creating Configuration Profiles

You can use the iPhone Configuration Utility to build out a configuration profile (or provisioning profile) and then upload it to the JSS. If you use the Casper portal, there is no need to bounce between the iPhone Configuration Utility and the web browser. You can substitute variables and have the username that the user uses to authenticate against the directory service in order to populate fields. For example, in an Exchange server environment you might want to substitute a variable instead of actual usernames. We spent a considerable amount of time in Chapter 8 covering the iPhone Configuration Utility. That time was well spent because JAMF matched its settings to those of the iPhone Configuration Utility, meaning that the options are similar if not identical.

To create a new profile, click the Mobile Device Profiles option under the Management tab. You will see a list of all of the profiles that you have created (Figure 9–22). You can edit and delete those that you have already created. To add a new profile, click the Add Profile button.

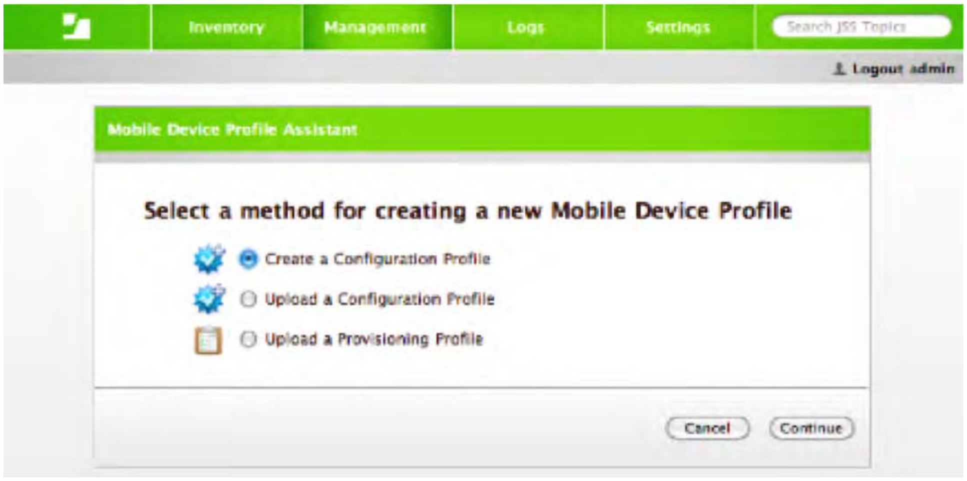


Configuration Profiles

	Display Name	Description	Scope		
●	asdf		0|0|0|0|0	Edit	Delete
●	Exchange Demo		0|0|0|0|0	Edit	Delete
●	testfasfasfasfasd		0|0|0|0|0	Edit	Delete

Figure 9–22. *Listing mobile device profiles*

You will then see a screen allowing you to create a configuration profile or to upload a configuration or provisioning profile. Select the Create a Configuration Profile button and then click the Continue button, as shown in Figure 9–23.

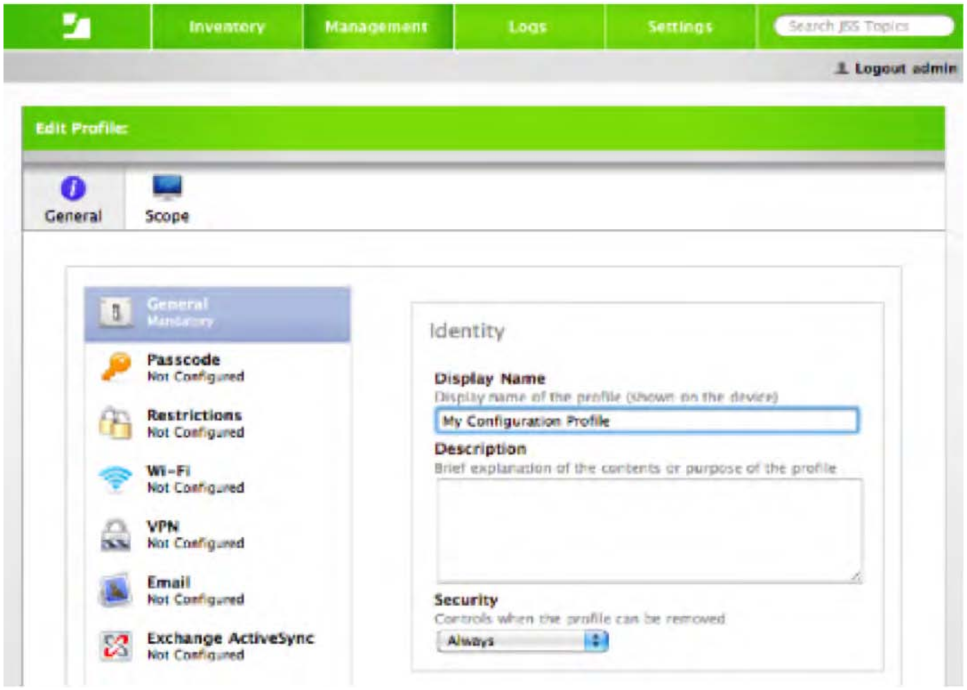


Select a method for creating a new Mobile Device Profile

[✓ Create a Configuration Profile]
Upload a Configuration Profile
Upload a Provisioning Profile

Figure 9–23. *Choosing the type of profile*

At the General tab, you will see a screen similar to the iPhone Configuration Utility. Provide a Display Name for the profile and optionally a description (Figure 9–24). The settings for most of these options are the same as those described in Chapter 8.



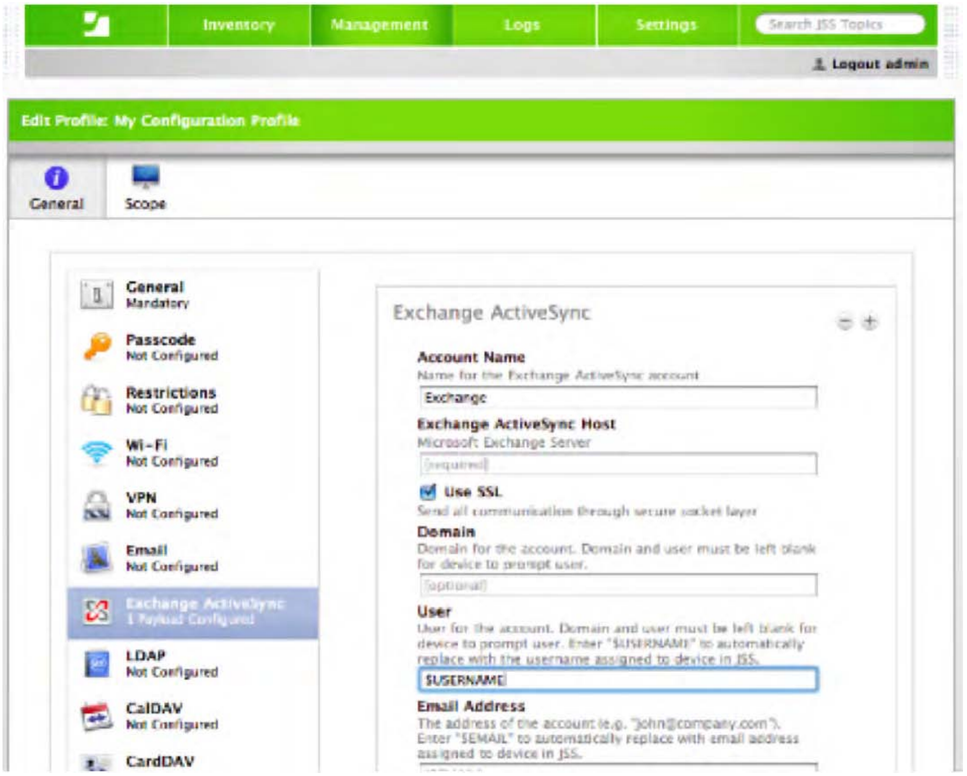
Display Name
Display name of the profile (shown on the device)
My Configuration Profile

Description
Brief explanation of the contents or purpose of the profile

Security
Controls when the profile can be removed
Always

Figure 9–24. *Naming the Profile*

Next, click the Exchange ActiveSync option. You will then see the standard options, again similar to those covered in Chapter 8. But in this case, you have the option to provide a variable rather than the user's actual username. You can use the `$USERNAME` variable (as seen in Figure 9–25). You can also use the `$EMAIL` variable for the e-mail address of the user being configured. These options allow you to deploy fully functional settings (for example, fully configured mail accounts) to users over the air without touching devices.



Exchange ActiveSync

Account Name
Name for the Exchange ActiveSync account
Exchange

Exchange ActiveSync Host
Microsoft Exchange Server
(required)

[✓] **Use SSL**
Send all communication through secure socket layer

Domain
Domain for the account. Domain and user must be left blank for device to prompt user.
(optional)

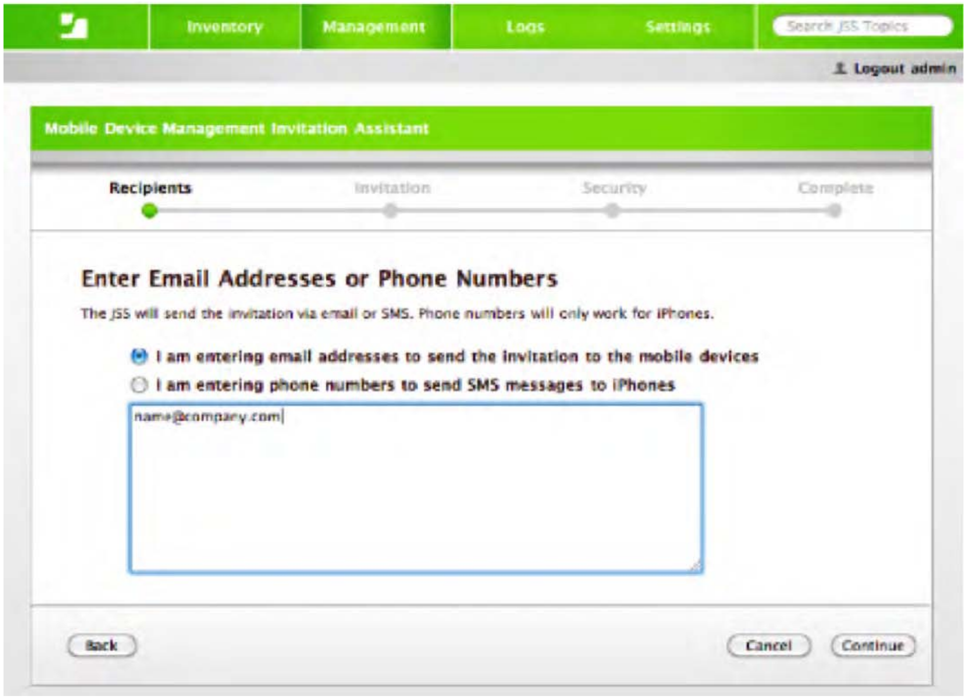
User
User for the account. Domain and user must be left blank for device to prompt user. Enter "$USERNAME" to automatically replace with the username assigned to device in JSS.
$USERNAME

Email Address
The address of the account (e.g. "john@company.com"). Enter "$EMAIL" to automatically replace with email address assigned to device in JSS.

Figure 9–25. *Using variables in configuration profiles*

Enrolling Devices

You can't push anything to a device until you enroll it. Luckily, this is a process similar among all of the vendors who currently inhabit the market for iOS management. To send an invitation from the JSS, log in and click the Management tab. Then click the Mobile Over-the-Air Enrollment link. You will see the Recipients step in the Mobile Device Management Invitation Wizard. As indicated in Figure 9–26, you can choose to send invitations through mail or SMS. If you choose to send through mail, users will need a valid e-mail account already configured on the device in order to accept your invitation, or access to web mail to click the link. This option is usually best in environments where you have devices already deployed and do not have the UDID documented for the device. This option is also necessary for iPod Touch and iPads without a cell plan attached to them. Alternatively, you can use the SMS feature, which is best for large deployments of new devices, when you have the SMS information is accessible for each device.



Mobile Device Management Invitation Assistant

Enter Email Addresses or Phone Numbers

The JSS will send the invitation via email or SMS. Phone numbers will only work for iPhones.

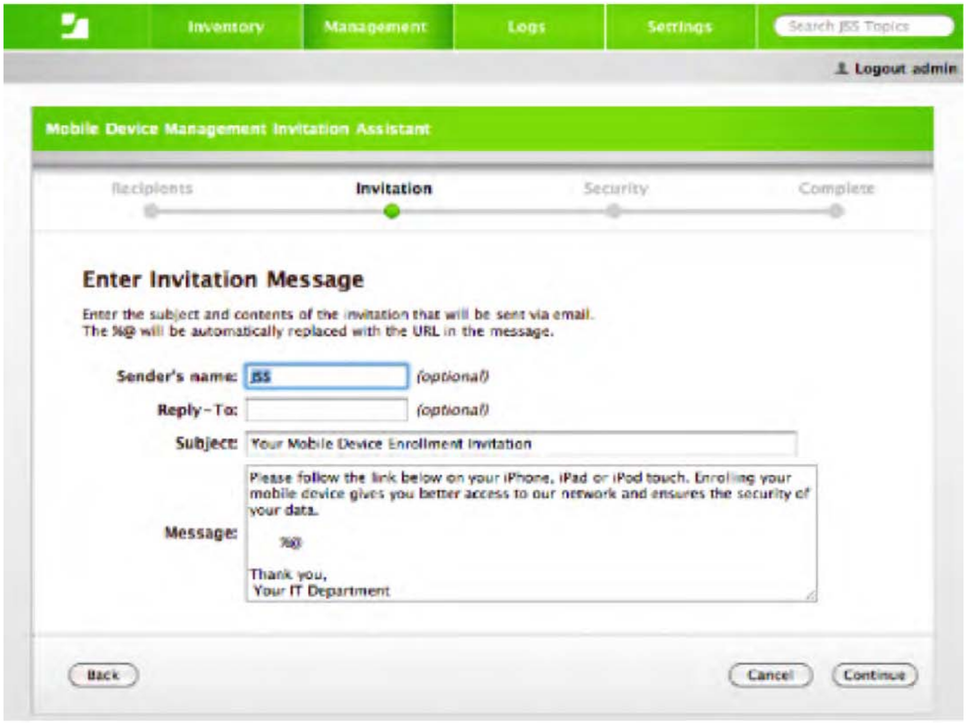
[✓ I am entering email addresses to send the invitation to the mobile devices]

I am entering phone numbers to send SMS messages to iPhones

name@company.com

Figure 9–26. *Defining invitation recipients*

After you have provided the e-mail addresses or phone numbers to send SMS messages to, click the Continue button. You will see the Invitation step of the assistant (Figure 9–27). On this screen, you can configure two optional settings: the name the user will see as the message's sender and the Reply-To information, indicating the address two which a response will be sent. You can also indicate the Subject of the message and write the body of the message itself (note the inclusion of the `%@` variable in the body that expands into the link to tap for enrollment). The body can be a useful place to store instructions, explain why you are sending the message, and indicate under whose authority the enrollment is being performed.



Enter Invitation Message

Enter the subject and contents of the invitation that will be sent via email.
The %@ will be automatically replaced with the URL in the message.

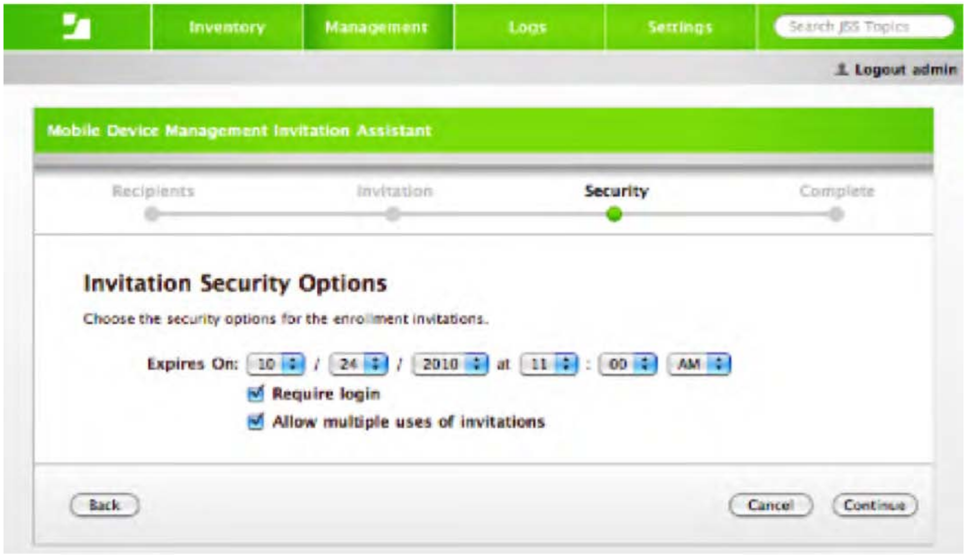
Sender's name: JSS (optional)
Reply-To: (optional)
Subject: Your Mobile Device Enrollment Invitation
Message: Please follow the link below on your iPhone, iPad or iPod touch. Enrolling your mobile device gives you better access to our network and ensures the security of your data.

%@

Thank you,
Your IT Department

Figure 9–27. *Crafting the invitation communication*

When you are satisfied with your invitation customizations, click the Continue button. You will then set security options for the invitation you are about to send out. As Figure 9–28 shows, you can configure the expiration date and time, indicate whether the user must authenticate by using a known good password (that is, from Active Directory or Open Directory), and set whether a single invitation can be used on multiple devices. The Allow Multiple Uses of Invitations option allows an organization to provide users with iPad, iPod Touch, and iPhones with one e-mail that the users can then use on multiple devices. If your users do not have more than one device, then, for security purposes, it is recommended to disable this option.



Invitation Security Options

Choose the security options for the enrollment invitations.

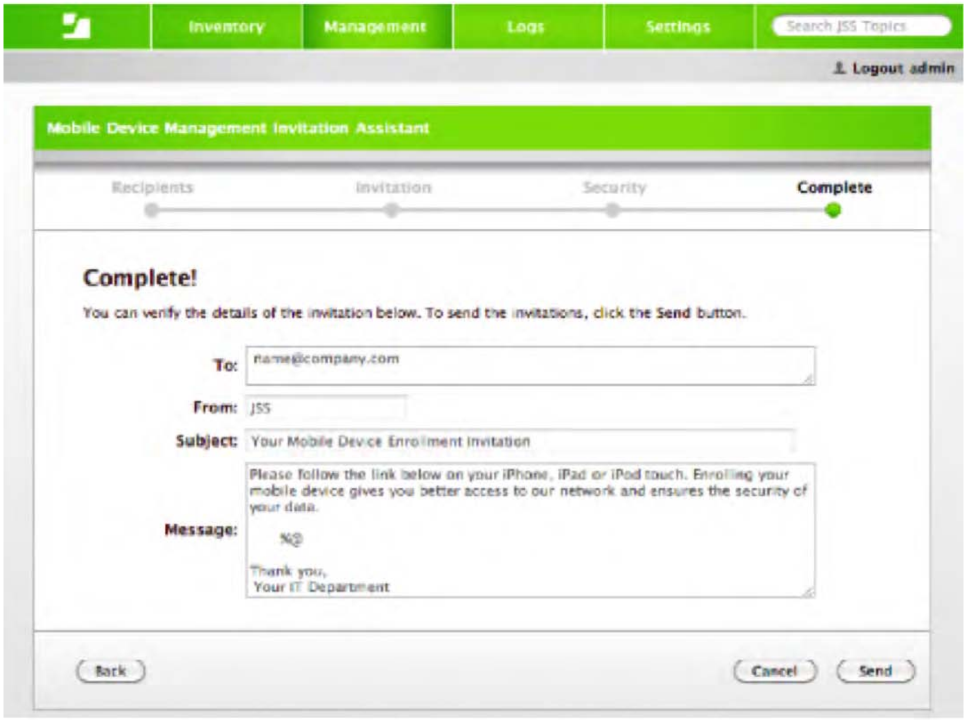
Expires On: 10 / 24 / 2010 at 11 : 00 AM

☑ Require login

☑ Allow multiple uses of invitations

Figure 9–28. *Securing the option to enroll*

When you are finished customizing the security options for your invitation, click Continue. At the final step of the assistant, review the details of the message that will be sent out, which can be seen in Figure 9–29. Then click the Send button and select a device to verify that the invitation was delivered.



Complete!

You can verify the details of the invitation below. To send the invitations, click the **Send** button.

To: name@company.com

From: JSS

Subject: Your Mobile Device Enrollment Invitation

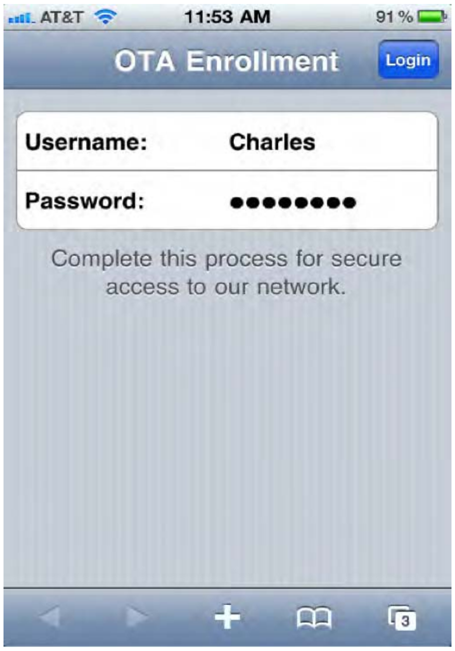
Message: Please follow the link below on your iPhone, iPad or iPod touch. Enrolling your mobile device gives you better access to our network and ensures the security of your data.

%@

Thank you,
Your IT Department

Figure 9–29. *Sending the invitations to enroll*

On the iOS-based devices, you should then see an e-mail with the link. Tapping the link takes you to the OTA Enrollment page, where you provide the username and password for the LDAP environment, as indicated in Figure 9–30.



Username: Charles

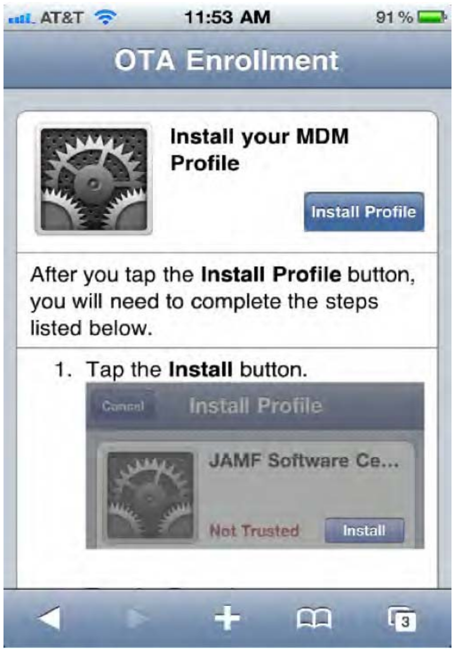
Password: ●●●●●●●●

Complete this process for secure access to our network.

Figure 9–30. *Authenticating for enrollment*

Enter that information and then tap the Login button to be taken to the MDM profile installation page (Figure 9–31). You will see a customizable page that tells the user how to install the profile (which can be done by tapping the Install Profile button).

NOTE: This page can be customized by using the global settings, as explained in the "Configuring Global Settings" section earlier in this chapter.


AT&T 11:53 AM 91 %

OTA Enrollment

Install your MDM Profile

Install Profile

After you tap the **Install Profile** button, you will need to complete the steps listed below.

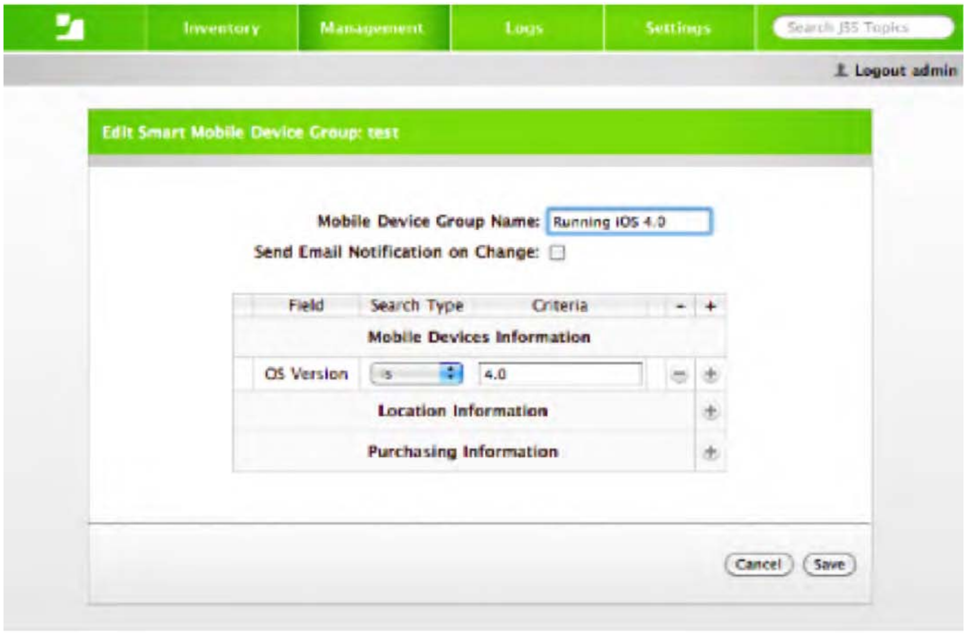
1. Tap the **Install** button.

Figure 9–31. *Enrolling devices*

The device is then enrolled into the server, having the MDM profile installed. That profile allows the MDM agent to communicate back to the server.

After you have enrolled a few devices, you will find that you need to manage them. The best way to do this en masse is through groups. At the Management tab, you can create what are known as *static groups* and *smart groups*. Smart groups are dynamically generated based on attributes (criteria) for the device, the location of the device, and the purchasing information pertaining to the device (for example, a warranty expiring within 30 days). Static groups require you to explicitly add a device to the group.

To create groups, from the Management tab click Smart Mobile Device Groups or Static Mobile Device Groups. Click the plus sign (+) to add new groups. If you are adding a smart group, you will see a screen similar to Figure 9–32, where you can specify more criteria for grouping devices that update dynamically.



Mobile Device Group Name: Running iOS 4.0
Send Email Notification on Change: ☐

Field	Search Type	Criteria
	Mobile Devices Information	
OS Version	is	4.0
	Location Information	
	Purchasing Information	

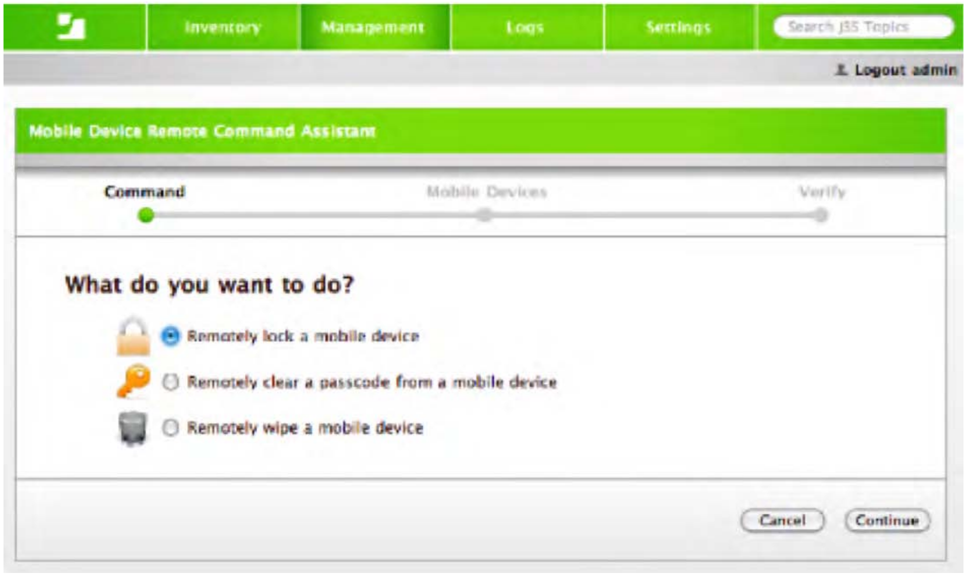
Figure 9–32. *Selection criteria for smart groups*

NOTE: You can import devices from a CSV file by using code available through the resource kit from JAMF.

Managing Devices

After your devices have been enrolled into the JSS, you can then send commands to them to perform specific tasks. At the time of this writing, three main commands are available: remote wipe, remote passcode reset, and remote lock (which locks a device until a specified passcode is entered). Because you are telling a device that is remote to do something immediately (run a command), these are referred to as *remote commands*.

To access the remote commands, start at the Management tab and then click Remote Commands. On the Remote Commands screen, you will see the option for each of the aforementioned commands that can be run (Figure 9–33).

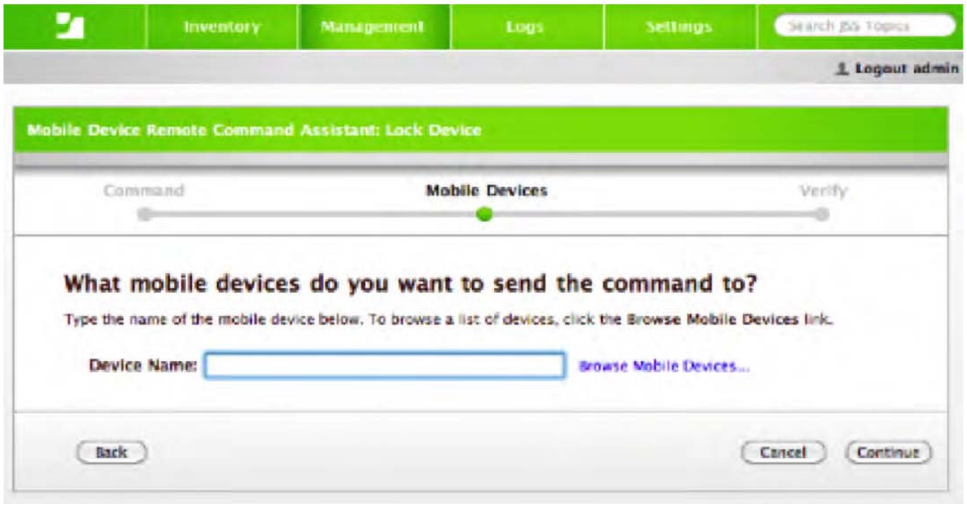


What do you want to do?

- [✓ Remotely lock a mobile device]
- Remotely clear a passcode from a mobile device
- Remotely wipe a mobile device

Figure 9–33. *Choosing a security command*

Click the option for the command you would like to run and then click the Continue button. You are then prompted to provide the device name, as shown in Figure 9–34. You can enter the name in one of three ways: begin typing the device name and then select from a list of devices that match what you are typing, type the full name, or click the Browse Mobile Devices button for a list of devices from which to select.



What mobile devices do you want to send the command to?

Type the name of the mobile device below. To browse a list of devices, click the Browse Mobile Devices link.

Device Name:

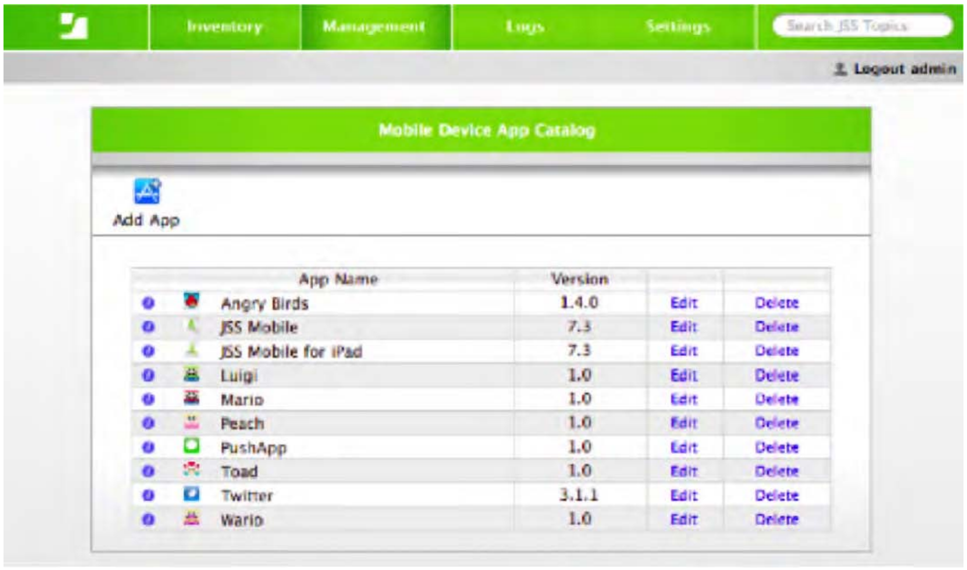
Figure 9–34. *Choosing a device to wipe (Choose carefully!)*

After you have selected the device to perform the action on, click the Continue button and verify that you would like to perform the remote command. These remote commands enable you to keep the devices secure and under your organization's control. As you saw in Chapter 4, your users can also remotely wipe a device by using Microsoft Exchange or Find My iPhone (for MobileMe users), but this provides a way for the administrator to perform these same tasks from a centralized location.

Adding Applications to the Catalog

The JAMF Software Server can also be used to deploy applications to the device. You cannot at this time push an application to the device by using the MDM APIs for iOS. However, you can provide what can be considered a private App Store. The Casper Suite does this through what is known as an App Catalog. To use the feature, you will add applications into the App Catalog on the JSS and then use the Self Service portal to install them from the device.

To add an application to the App Catalog, go to the Management tab and then click the link for Mobile Device App Catalog. The catalog is shown in Figure 9–35. From here, you will see a list of the applications in the App catalog, and you can Edit them (for example, change the icon or update the application to a new IPA file) or Delete them (that is, remove them from the Self Service portal).

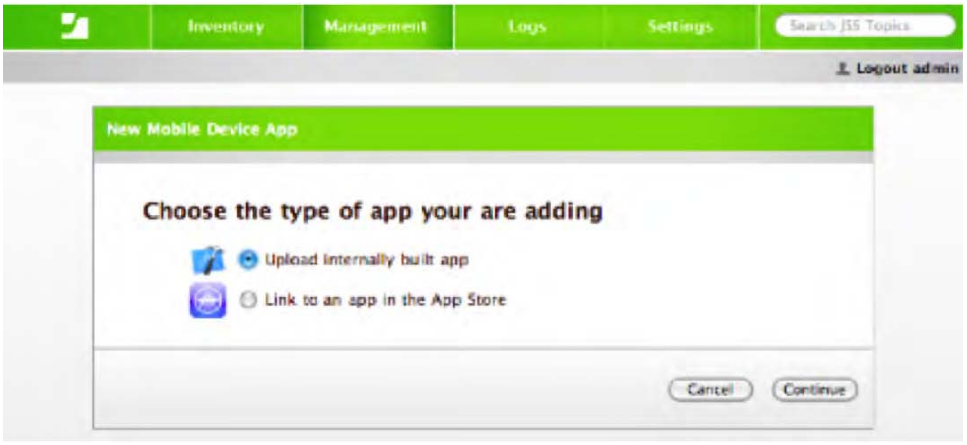


App Name	Version		
Angry Birds	1.4.0	Edit	Delete
JSS Mobile	7.3	Edit	Delete
JSS Mobile for iPad	7.3	Edit	Delete
Luigi	1.0	Edit	Delete
Mario	1.0	Edit	Delete
Peach	1.0	Edit	Delete
PushApp	1.0	Edit	Delete
Toad	1.0	Edit	Delete
Twitter	3.1.1	Edit	Delete
Wario	1.0	Edit	Delete

Figure 9–35. *The App Device Catalog*

Click the Add App icon to start the process of adding a New Mobile Device App (using the appropriately named New Mobile Device App screen shown in Figure 9–36). The Mobile Device App can be an application from the Apple App Store or a compiled application from Xcode in the form of an IPA file (using the Build and Archive option). For this example, we will use an internally built app (thus a compiled Xcode project).

Provided that you have a project (see Chapter 7 for more information on building projects), click Upload Internally Built App and then click Continue.



Choose the type of app your are adding

[✓ Upload internally built app]
Link to an app in the App Store

Figure 9–36. *Adding applications to the catalog*

You will then see options for the application being installed. The Edit Mobile Device App screen (Figure 9–37) has each of the options. Fill in the following fields:

App Name: The name that the application will have after it is installed on the device.

Bundle ID: Can match the bundle ID of the application from the Xcode project.

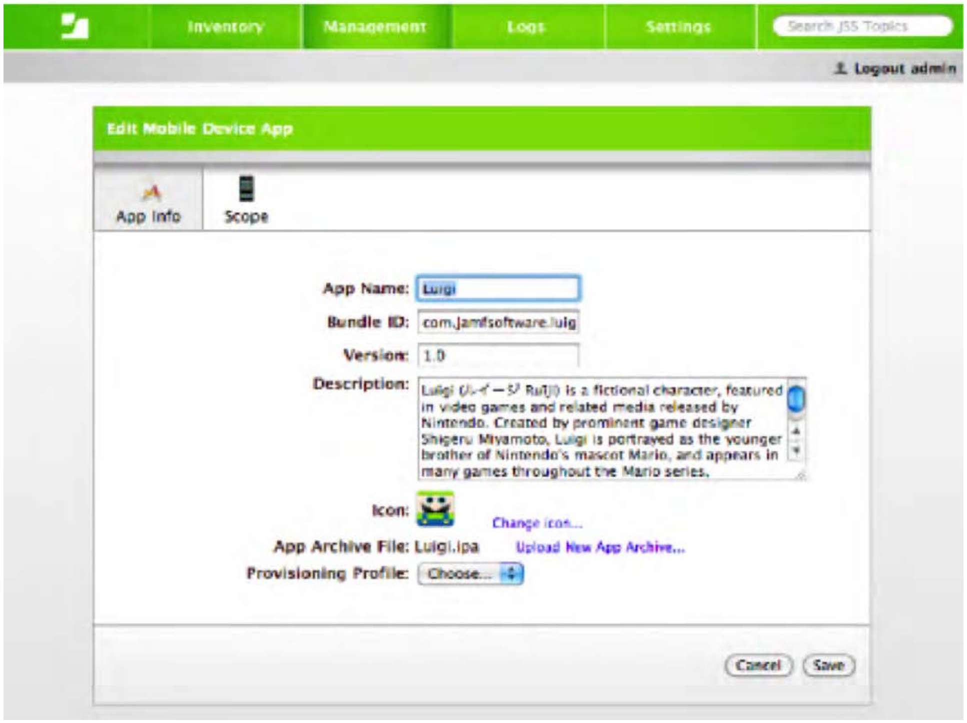
Version: This is a text field you provide that is not linked to the Xcode project.

Description: A text description to be shown to the user, explaining what the application is for.

Icon: The icon that will be used on the screen of the iOS device to access the application.

App Archive File: The IPA archive file from Xcode.

Provisioning Profile: The profile to which the application is attached.

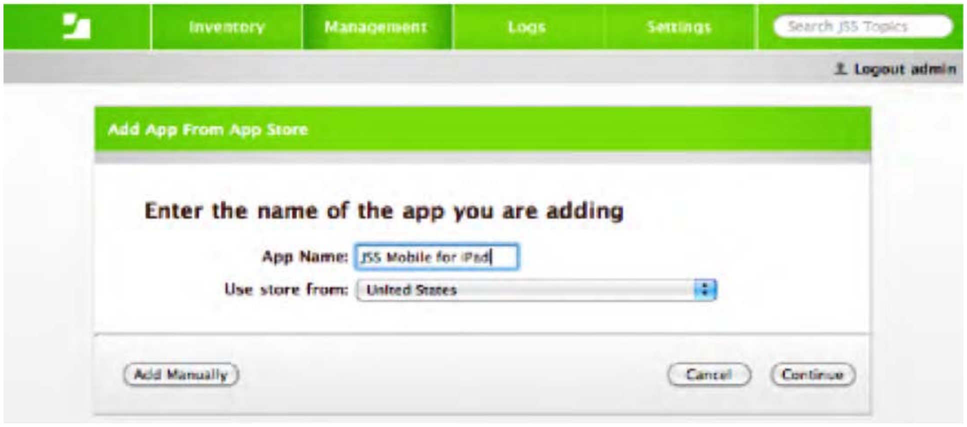


App Name: Luigi
Bundle ID: com.jamfsoftware.luig
Version: 1.0
Description: Luigi (ルイージ Ruīji) is a fictional character, featured in video games and related media released by Nintendo. Created by prominent game designer Shigeru Miyamoto, Luigi is portrayed as the younger brother of Nintendo's mascot Mario, and appears in many games throughout the Mario series.
Icon:
App Archive File: Luigi.ipa
Provisioning Profile: Choose...

Figure 9–37. *Providing application information for internally built apps*

After you are satisfied with your options, click the Save button. You can then click the Scope icon to assign which groups will be allowed to access, and therefore install, the application.

If you instead are providing an application from Apple's App Store, you will have the option to add that by choosing the Link to an App in the App Store option. This is akin to a list of recommended (and therefore, approved) applications. At the Add App from App Store screen, you provide a name for the application being added to the catalog and indicate the country where the application's App Store is located (Figure 9–38).



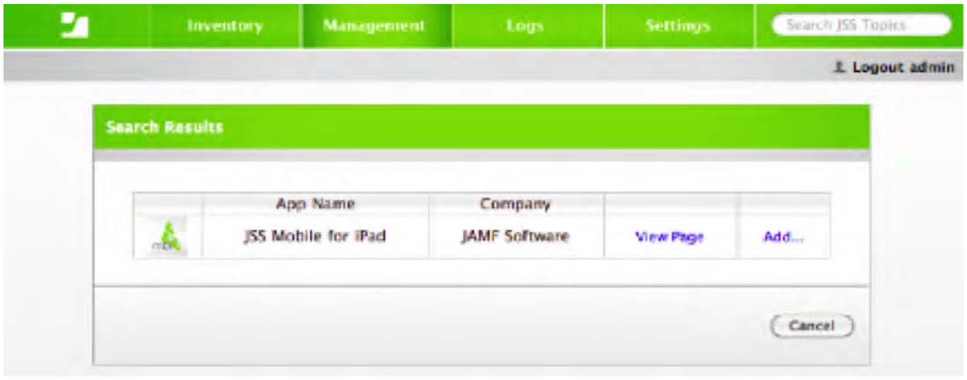
Enter the name of the app you are adding

App Name: JSS Mobile for iPad

Use store from: United States

Figure 9–38. *Finding the app store application to add*

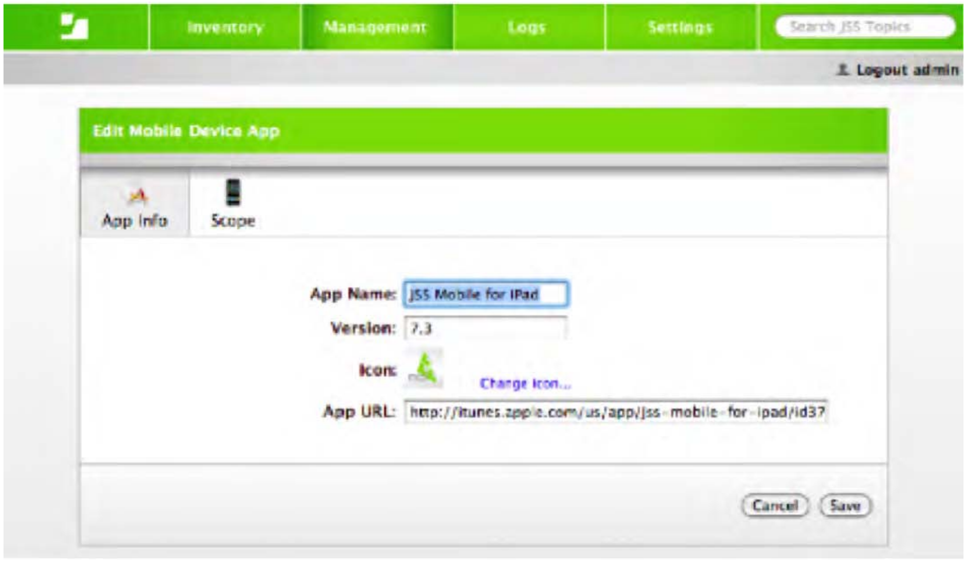
Click the Continue button to search for the application in question. You will see a Search Results page with each item that matched your search, as shown in Figure 9–39. You can then view the page of the application to make sure it is the one you are looking for, or simply click the Add button to add it.



	App Name	Company		
	JSS Mobile for iPad	JAMF Software	View Page	Add...

Figure 9–39. *Search results from Apple's App Store*

After you click on the Add button, you will see another rendition of the Edit Mobile Device App screen. However, as shown in Figure 9–40, this one shows only the URL used to access the application. (This URL can be obtained manually from iTunes by right-clicking or Control+clicking an application and then clicking Copy Link). The name the application shows on the iOS screen, the version number, and the icon can be changed here, and the scope options are similar to those from internally built applications. When you are satisfied with your options, click Save to publish the application to the groups defined in the Scope section for the application.



App Name: JSS Mobile for iPad
Version: 7.3
Icon: Change Icon...
App URL: http://itunes.apple.com/us/app/jss-mobile-for-ipad/id37

Figure 9–40. *Setting up the Mobile Device App for App Store URLs*

Applications can then be accessed by using the Self Service option (via the Self Service web clip enabled in the Global Settings) deployed at enrollment time (if that option was selected).

Providing Self-Service

One of the more complicated tasks facing systems administrators looking to mass deploy to iOS is how to push out applications. In the last major release of the Casper Suite, JAMF took the idea of the App Store and made a self-servicing portal for Mac OS X. Through this portal, users could install software without an administrative username and password on the user's Mac OS X computer, provided the application was in the organization's self-service portal, a repository of software that was approved by the organization. In Casper 8.0, JAMF has added this same feature for the iPhone, which addresses many of the challenges for deploying applications to iOS-based devices that have been discussed throughout this book.

When a device has been enrolled into the JAMF Software Server environment, you will see an application on the iOS-based device called Self Service. Tapping Self Service shows the users a list of software that is approved and/or recommended by the software administrator, as shown in Figure 9–41. The users can then tap each application and install the ones they wish to have, or they can tap the Install All button to install all of the software titles concurrently. The applications are divided into three categories along the bottom of the screen, with new Internal Apps, App Store Apps and Updates (from left to right). Internal Apps being applications distributed only within your organization, App Store Apps being links to applications hosted on Apple's App Store and Updates being applications that have updates available to them on either.

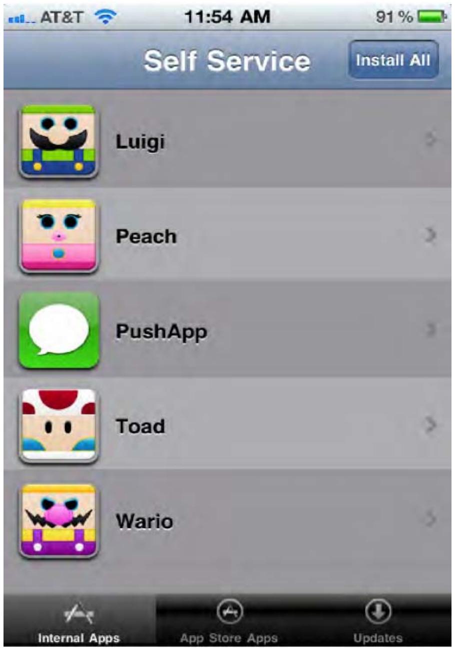


Figure 9–41. *Internal apps in Self Service*

The ability to install software through a portal is a great feature for any environment, but the ability to install all of the apps concurrently is a must for environments with multiple apps being distributed to all of your devices.

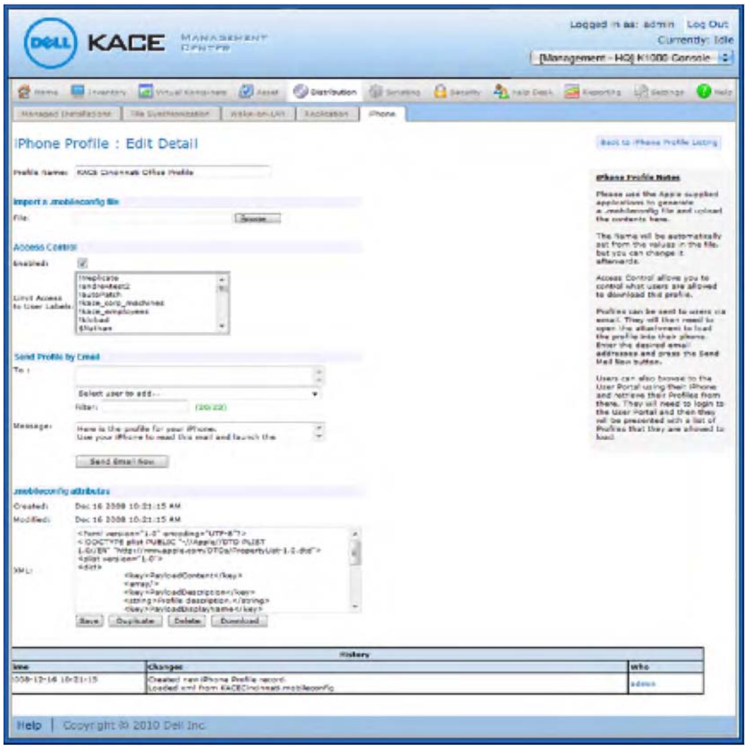
KACE Appliances

The KACE Deployment and Management appliances, from Dell, can be leveraged to provide centralized configuration management of the iPhone and iPod Touch. KACE appliances can be leveraged to provision, configure, and control policies with more granularity than can be found with the iPhone Configuration Utility. For example, you can leverage groups with your policies, monitor utilization and application installations, and track plans and renewals for wireless contracts for the iPhone (Figure 9–42). You can also use KACE appliances for the following:

- Performing iPhone configuration profile management
- Assigning profiles to user groups—for example, by function or geography
- Distributing profiles by e-mail
- Downloading profiles available from the K1000 or K2000 user portal
- Enabling iPhone user access to the K1000 or K2000 user portal

- Tracking and reporting iPhone asset information
- Tracking individual iPhone information
- Tracking the number of iPhone users
- Tracking iPhone equipment versions
- Tracking assigned serial numbers, including IMEI and ICCID identifiers
- Monitoring installed applications
- Allowing access to corresponding contracts
- Tracking individual plans and renewal dates
- Keeping asset data current via scheduled iPhone K-script refresh
- Delivering e-mail alerts based on changes in the asset fields
- Tracking related dependencies, that is, carrier contracts and associated renewal dates

More on KACE appliances and the ability to manage iPhone and iPod touch can be found at `http://www.kace.com/products/systems-management-appliance/features/iphone-management.php`



iPhone Profile : Edit Detail

Profile Name: KACE Cincinnati Office Profile

Import a .mobileconfig file

File:

Access Control

Enabled: ✓

Limit Access to User Labels: !!replicate, !andrewtest2, !autoPatch, !kace_corp_machines, !kace_employees, !kbload, !Nathan

Send Profile by Email

To :

Select user to add--

Filter: (20/22)

Message: Here is the profile for your iPhone. Use your iPhone to read this mail and launch the

.mobileconfig attributes

Created: Dec 16 2008 10:21:15 AM

Modified: Dec 16 2008 10:21:15 AM

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST
1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>PayloadContent</key>
        <array/>
        <key>PayloadDescription</key>
        <string>Profile description.</string>
        <key>PayloadDisplayName</key>
```

iPhone Profile Notes

Please use the Apple supplied applications to generate a .mobileconfig file and upload the contents here.

The Name will be automatically set from the values in the file, but you can change it afterwards.

Access Control allows you to control what users are allowed to download this profile.

Profiles can be sent to users via email. They will then need to open the attachment to load the profile into their phone. Enter the desired email addresses and press the Send Mail Now button.

Users can also browse to the User Portal using their iPhone and retrieve their Profiles from there. They will need to login to the User Portal and then they will be presented with a list of Profiles that they are allowed to load.

History

Time	Changes	Who
2008-12-16 10:21:15	Created new iPhone Profile record. Loaded xml from KACECincinnati.mobileconfig	admin

Figure 9–42. *The KACE Management Center*

NOTE: KACE was recently acquired by Dell Computers.

MobileIron

MobileIron provides another solution, similar to other deployment tools. MobileIron for iOS includes a corresponding iPhone application, available from the App Store. You can see how the MobileIron application appears to end users in Figure 9–43. With MobileIron, you can leverage a catalog of recommended applications that users are able to download through the App Store, similar to the Self Service tool from the Casper Suite. Because MobileIron uses an installed agent on the device (Figure 9–43), it is also able to track dropped calls and track locations of devices, which could be used for breadcrumb mapping, or tracking the devices movements based on GPS coordinates.

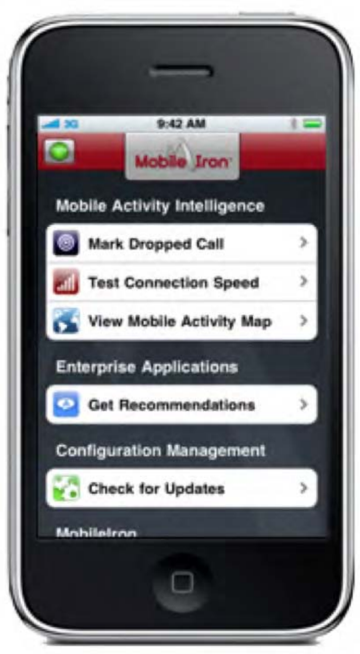


Figure 9–43. *MobileIron at work*

NOTE: MobileIron can also test the connection speed of an iOS-based device.

As one of the early developers of tools for the mass management of the iOS market, MobileIron has also released a tool called MobileIron Sentry for iPad. Sentry uses the iPad to provide IT with visibility and control over various aspects of ActiveSync in your Exchange environment. Using Sentry (Figure 9–44), you can see statistics about your mobile deployment via an easy-to-read dashboard, helpful when tracking the health of your environment.

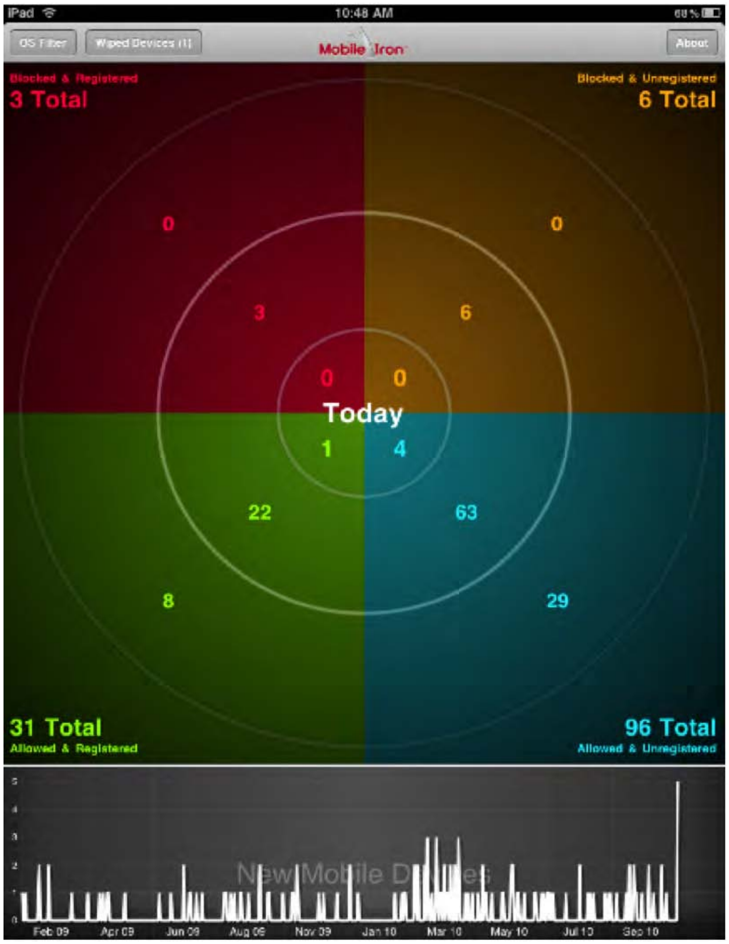
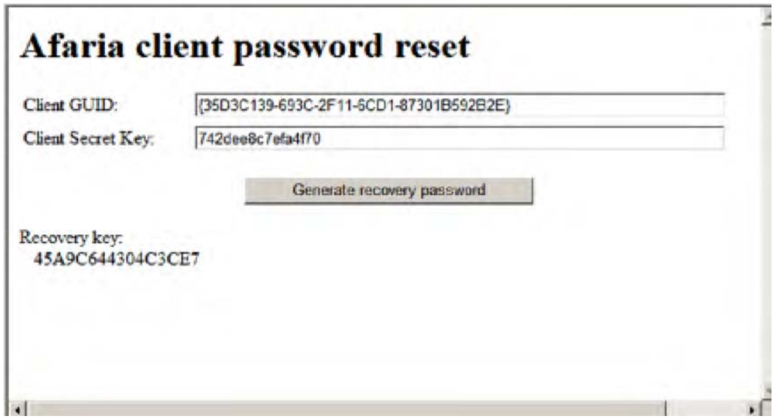


Figure 9–44. *MobileIron's Sentry*

Sybase Afaria

Many an enterprise has been using Sybase products for decades. That trust of the popular Sybase database engine seems to extend into the capabilities to manage numerous mobile devices by using a single, unified console in their product Afaria. Among the supported platforms that Afaria can manage is the iPhone.

As with many solutions that support numerous platforms, Afaria does not have a lot of specialty features for the iPhone. Instead it has a very legitimate presence in the market and a host of other products from Sybase, including Mobile Sales and iAnywhere Mobile Office for the iPhone, that provide a more comprehensive suite of products than just management en masse. Afaria includes support for most of the basic functions of MobileIron, TARMAC, and others with added support for features meant to specifically reduce support needs of mobile device users, such as resetting passwords (Figure 9–45), enrolling new previously unknown devices, and so forth.



Afaria client password reset

Client GUID: {35D3C139-693C-2F11-6CD1-87301B592B2E}

Client Secret Key: 742dee8c7efa4f70

Generate recovery password

Recovery key:
45A9C644304C3CE7

Figure 9–45. *Resetting a password with Afaria*

TARMAC

The final tool that I will reference for deployment of iOS-based devices is TARMAC. TARMAC configuration is done within the TARMAC web portal (Figure 9–46). Here you can import profiles, associate profiles with devices, integrate with your organization's directory services, and push out patch management policies.

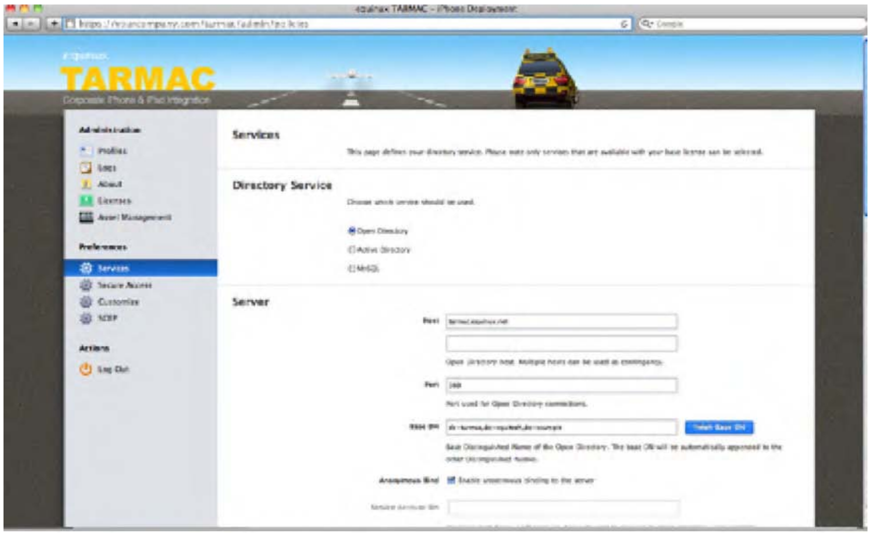


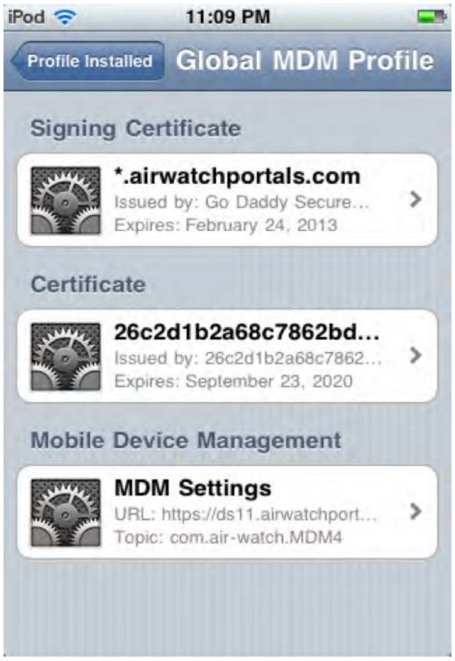
Figure 9–46. *Configuring TARMAC*

After the server has been properly configured, TARMAC then allows a device to authenticate to a web portal, which generates the configuration for the specified user at the time of login. TARMAC is one of the easier solutions available, following very closely along Apple's example code that was referenced in the beginning of this chapter.

Removing the Profiles

During all of this testing, there's a good chance that you may have installed a whole slew of profiles on your device that you will want to get rid of. Or if a device is leaving the management environment of your organization, you may want to remove the MDM, configuration, or provisioning profiles you deployed to the devices. Therefore, you will eventually want to remove a profile.

Profiles can be removed quickly and easily on your device in most cases without a reset. To delete a profile, first open the Settings application on the iOS-based device. Then tap General Settings and then Profiles. You will see a list of all of the profiles that have been installed on the device (Figure 9–47).



Signing Certificate

*.airwatchportals.com
Issued by: Go Daddy Secure...
Expires: February 24, 2013

Certificate

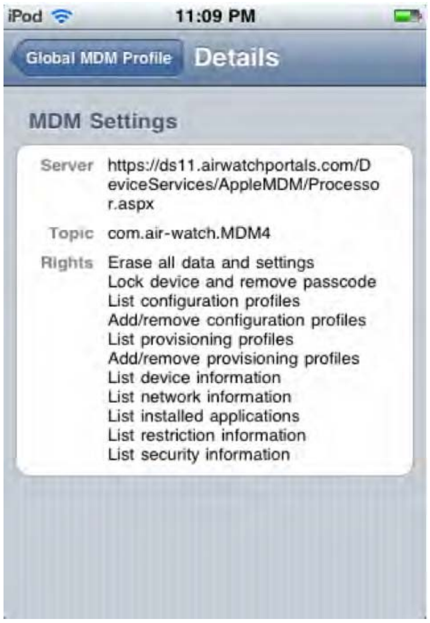
26c2d1b2a68c7862bd...
Issued by: 26c2d1b2a68c7862...
Expires: September 23, 2020

Mobile Device Management

MDM Settings
URL: https://ds11.airwatchport...
Topic: com.air-watch.MDM4

Figure 9–47. *Installed profiles*

Tap on a profile to see the Details, which will show the rights that the profile has when managing the device (Figure 9–48).

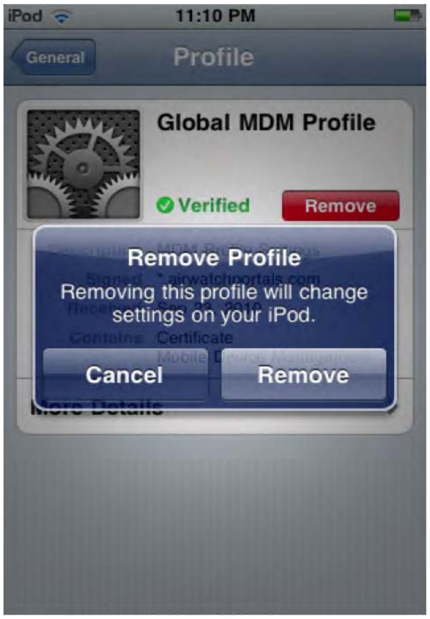


MDM Settings

Server	https://ds11.airwatchportals.com/DeviceServices/AppleMDM/Processor.aspx
Topic	com.air-watch.MDM4
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

Figure 9–48. *Showing profile permissions*

To remove the profile, tap the Remove button on the Profile screen. You will then see a screen requiring you to confirm the removal of the profile, as shown in Figure 9–49. Click Remove to complete the process.



Remove Profile

Removing this profile will change settings on your iPod.

Figure 9–49. *Removing the Profile*

After the profile is removed, the device can no longer be managed with MDM policies deployed from the server. Practically any third-party product that can push a profile to a device can also revoke it, so you can also centrally remove these MDM policies.

Summary

One of the most critical aspects of mass integration of iOS in my organizations is the act of actually deploying profiles. Starting with the profiles created in Chapter 8, and using the tools and techniques in this chapter, we have laid out a framework that can be used to deploy hundreds or even tens of thousands of devices.

Most organizations that are going to deploy iOS-based devices en masse will need a strategy for automated deployment. To aid you in your endeavor, there are several competing products, many of which are fairly new to the market. Whether you think that AirWatch, the Casper Suite, KACE, MobileIron, or TARMAC are for you after reading this chapter, you should make sure to test each, reviewing their impact to the budget of your project and the features that you get as a return for that investment. Once you have found a product that meets your criteria for deployment, the rest of the pieces will start to fall in place.

Now that you've looked at many of the technical hurdles that you are likely to face in the first nine chapters of this book, we're going to move on to something a little more fun. In a look at many of the more enterprise-oriented applications available today for the iOS-based devices (Chapter 10), we will look at the true goal of any device in most environments: productivity.