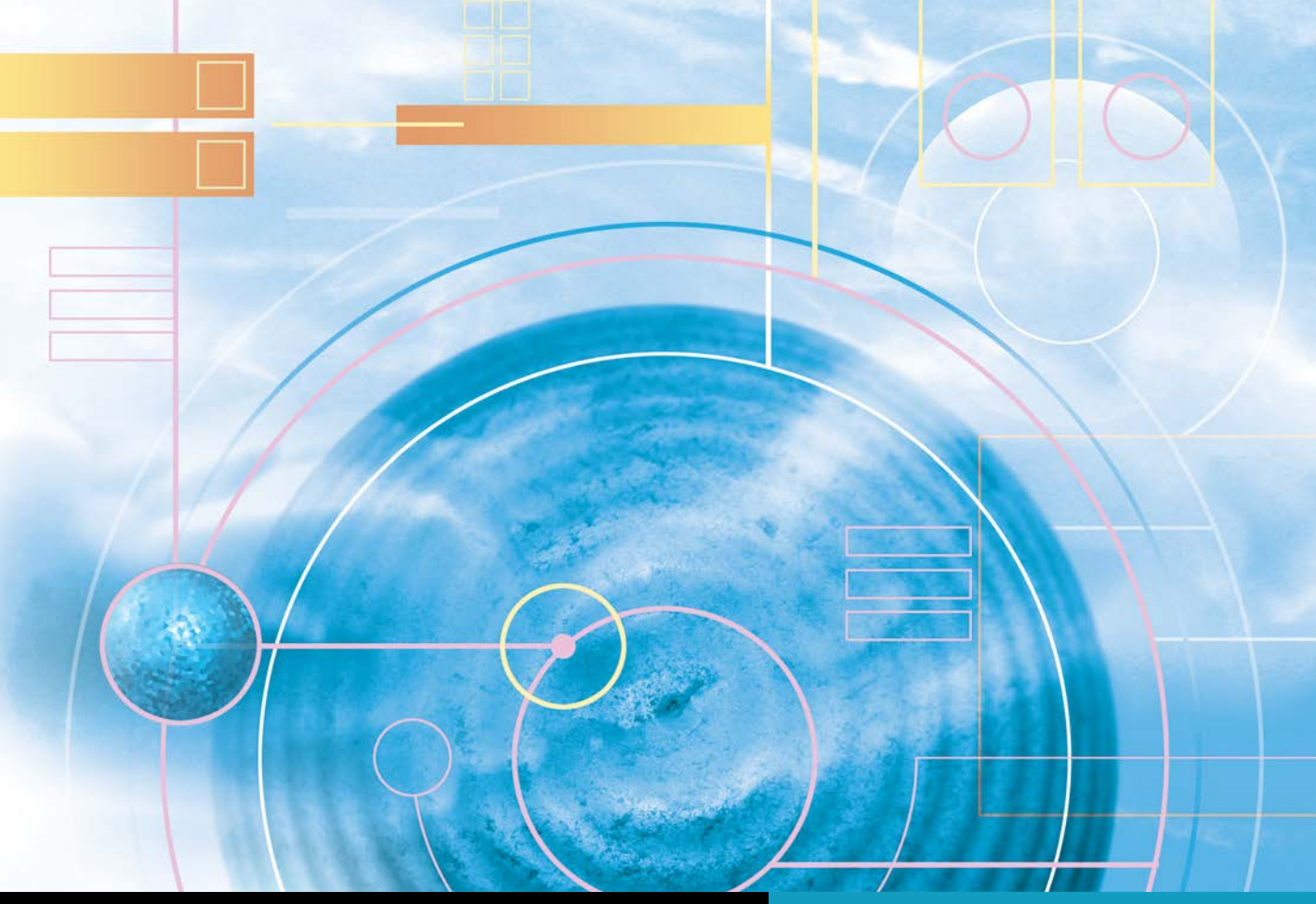


Apple Pro Training Series

OS X Server Essentials 10.10

Using and Supporting OS X Server on Yosemite

Arek Dreyer and Ben Greisler

Apple Certified

Certification Exam Preparation For
Apple Certified Technical Coordinator 10.10

Lesson and media files available for download

Apple Pro Training Series

OS X Server Essentials 10.10

Arek Dreyer and Ben Greisler

Apple Pro Training Series: OS X Server Essentials 10.10
Arek Dreyer and Ben Greisler

Peachpit Press
www.peachpit.com

To report errors, please send a note to *errata@peachpit.com*. Peachpit Press is a division of Pearson Education.

Apple Series Editor: Lisa McClain
Production Editor: Tracey Croom
Technical Editor: Adam Karneboge
Apple Reviewer: Susan Najour
Apple Project Manager: Debra Otterstetter
Copy Editor: Kim Wimpsett
Proofreader: Darren Meiss
Production Services: Happenstance Type-O-Rama
Indexer: Jack Lewis
Cover Illustration: Paul Mavrides
Cover Production: Happenstance Type-O-Rama

IMPORTANT: Some of the exercises contained in this guide can be temporarily disruptive, and some exercises, if performed incorrectly, could result in data loss or damage to system files. As such, it's recommended that you perform these exercises on a Mac computer that is not critical to your daily productivity.

ISBN 13: 978-0-13-403350-1
ISBN 10: 0-13-403350-7
9 8 7 6 5 4 3 2 1 Printed and bound in the United States of America
Alternate Binding:
ISBN 13: 978-0-13-403417-1
ISBN 10: 0-13-403417-1
9 8 7 6 5 4 3 2 1 Printed and bound in the United States of America

Thanks to my lovely wife, Heather Jagman, for her cheerful support.

—Arek Dreyer

My love and appreciation to my wife, Ronit, and my children, Galee and Noam, for their continued support through this project.

—Ben Greisler

This page intentionally left blank

Acknowledgments With the memory of Steve Jobs still fresh in our minds, thank you to Tim Cook, Jonathan Ive, and everyone at Apple for continually innovating, surprising, and delighting customers.

Thank you to all the people who continue to help their users get the most out of OS X and iOS. Keep learning, and don't expect the pace of change to let up any time soon.

Thanks to the amazingly capable Lisa McClain, for gently making sure these materials made it into your hands, and to Scout Festa and Kim Wimpsett, for working their editorial and production magic.

Thanks to Adam Karneboge for adept corrections and suggestions.

Thanks to Schoun Regan for all his help.

Thank you, also, to the following people. Without your help, this book would be much less than what it is:

Mark Bulthaup	Scott George	Alby Rose
Craig Cohen	Charlie Heizer	John Signa
Gordon Davisson	Andre LaBranche	Cindy Waller
Weldon Dodd	Ben Levy	Simon Wheatley
Josh Durham	Tip Lovingood	Kevin White
Charles Edge	Jussi-Pekka Mantere	Josh Wisenbaker
Ed Faulkner	Sean Murphy	Eric Zelenka
Patrick Gallagher	Susan Najour	

Contents at a Glance

Bonus chapters mentioned in this eBook are available after the index.

See last page of this eBook for instructions on downloading your lesson files.

Table of Contents

Configuring Accounts

Managing Devices with Configuration Profiles

Sharing Files

Implementing Deployment Solutions

Using Collaborative Services

Bonus chapters mentioned in this eBook are available after the index.

See last page of this eBook for instructions on downloading your lesson files.

About This Guide

This guide serves as a tour of the breadth of functionality of OS X Server and the best methods for effectively supporting users of OS X Server systems. It is for both self-paced learners working independently and those participating in an instructor-led course. This guide is the curriculum for the Apple official training course Yosemite 201: OS X Server Essentials 10.10, a three-day, hands-on course that provides an in-depth exploration of how to configure and support OS X Server for Yosemite. This course is facilitated by an Apple Certified Trainer and is organized into multiple lessons, each containing instructor presentations followed by related student exercises.

GOALS

- Learn how this guide is organized to facilitate learning
- Set up an environment for self-paced exercises
- Introduce Apple Authorized Training and Certification

The primary goal of this guide is to prepare technical coordinators and entry-level system administrators for the tasks demanded of them by OS X Server; you will learn how to install and configure OS X Server to provide network-based services, such as configuration profile distribution and management, file sharing, authentication, and collaboration services. To help you become truly proficient, this guide covers the theory behind the tools you will use. For example, not only will you learn how to use Server app—the tool for managing services and accounts—but you will also learn about the ideas behind profile management, how to think about access to and control of resources, and how to set up and distribute profiles to support your environment.

You will learn to develop processes to help you understand and work with the complexity of your system as it grows. Even a single OS X Server computer can grow into a complicated system, and creating documentation and charts can help you develop processes so that additions and modifications can integrate harmoniously with your existing system.

This guide assumes you have some knowledge of OS X, because OS X Server is an app that you install on OS X (Yosemite). Therefore, you should be comfortable with basic navigation, troubleshooting, and networking in OS X. When working through this guide, a basic understanding and

knowledge of OS X is preferred, including knowledge of how to troubleshoot the operating system. Refer to *Apple Pro Training Series: OS X Support Essentials 10.10* from Peachpit Press if you need to develop a solid working knowledge of OS X.

> **NOTE ▶** Unless otherwise specified, all references to OS X refer to version 10.10 or later, and references to OS X Server refer to version 4.0, which at the time of this writing is the most current version available. Some screenshots, features, and procedures may be slightly different from those presented on these pages because of subsequent upgrades.

Learning Methodology

Each lesson in this guide is designed to give technical coordinators and entry-level system administrators the skills, tools, and knowledge to implement and maintain a network that uses OS X Server by doing the following:

- Providing knowledge of how OS X Server works
- Showing how to use configuration tools
- Explaining troubleshooting and procedures

The exercises contained within this guide are designed to let you explore and learn the tools necessary to manage OS X Server for Yosemite. They move along in a predictable fashion, starting with installing and setting up OS X Server and moving to more advanced topics such as performing multiprotocol file sharing, using access control lists, and permitting OS X Server to manage network accounts. It is required that you start from a Mac that is not yet running OS X Server and that you do not use this server as a production server.

This guide serves as an introduction to OS X Server and is not meant to be a definitive reference. Because OS X and OS X Server contain several open source initiatives and can be configured at the command line, it is impossible to include all the possibilities and permutations here. First-time users of OS X Server and users of other server operating systems who are migrating to OS X Server have the most to gain from this guide; still, others who are upgrading from previous versions of OS X Server will also find this guide a valuable resource.

OS X Server is by no means difficult to set up and configure, but how you use OS X Server should be planned in advance. Accordingly, this guide is divided into seven parts:

- Part 1, "Configuring and Monitoring OS X Server," covers planning, installation, initial configuration, and monitoring of OS X Server.
- Part 2, "Configuring Accounts," defines authentication and authorization, access control, and Open Directory and the vast functionality it can provide.
- Part 3, "Managing Devices with Configuration Profiles," covers managing devices with the Profile Manager service.
- Part 4, "Sharing Files," introduces the concept of sharing files over multiple protocols and controlling access to files with access control lists.
- Part 5, "Implementing Deployment Solutions," teaches you how to effectively use deployment services, NetInstall, the Caching service, and the Software Update service.
- Part 6, "Providing Network Services," introduces the network services, including Time Machine, VPN, DHCP, and Websites.
- Part 7, "Using Collaborative Services," focuses on setting up collaboration services together, starting with Mail, moving through Wiki, Calendar, and Contacts, and finishing with the Messages service.

Lesson Structure

Most lessons in this guide contain a reference section followed by an exercise section (the lessons on Caching and Software Update services do not contain exercises).

> **NOTE ▶** "Note" resources, like this one, offer important information to help clarify a subject. For example, some of the exercises in this guide may be disruptive. Consequently, it's recommended that you perform these exercises on an OS X computer that is not critical to your daily productivity.

The reference sections contain initial explanatory material that teaches essential concepts. The exercise sections augment your understanding of concepts and develop your skills through step-by-step instruction for both self-paced learners and the hands-on portions of an instructor-led course.

> **TIP** "Tip" resources, like this one, provide helpful hints, tricks, or shortcuts. For example, each lesson begins with an opening page that lists the learning goals and necessary resources for the lesson.

> **MORE INFO ▶** The "More Info" resources, like this one, provide ancillary information. These resources are merely for your edification and are not considered essential for the coursework.

Throughout this guide you'll find references to Apple Support articles. You can find these articles at the Apple Support website (www.apple.com/support), a free online resource containing the latest technical information for Apple products. We strongly encourage you to read the suggested articles and search the Apple Support website for answers to any problems you encounter.

We encourage you to explore two additional resources that Apple provides specifically for OS X Server: OS X Server Support (https://www.apple.com/support/osxserver/) and OS X Server: Advanced Administration (https://help.apple.com/advancedserveradmin/mac/4.0/).

Lesson files and bonus materials are available online when you redeem the access code supplied with your guide at www.peachpit.com/redeem. Detailed instructions for downloading files are provided later in this guide. Appendix A, "Lesson Review Questions & Answers," recaps each lesson through a series of questions that reinforce the material you learned in the guide. Try to answer each question yourself before looking at the answer. You can refer to various Apple resources, such as the Apple Support website and OS X Server documentation, as well as the lessons themselves to help you answer these questions. Appendix B, "Additional Resources," lists relevant Apple Support articles and recommended documents related to the topic of each lesson. You'll also find supplemental exercise material for exercises in Lesson 20 and Lesson 23. An "Updates & Errata" document will contain updates and corrections to the guide if any are available.

Exercise Setup

This guide is written so that both the self-paced learner and the attendee at an Apple Authorized Training Center (AATC) or Apple Authorized Training Center for Education (AATCE) can complete most of the exercises using the same techniques. Those attending Yosemite 201 at an AATC or AATCE will have the appropriate exercise setup provided as part of the training experience. Self-paced learners attempting these exercises will have to set up an appropriate environment using their own equipment.

NOTE ▶ Some of these exercises can be disruptive (for example, turning on the DHCP service may prevent devices on the local network from being able to browse Internet), and some exercises, if performed incorrectly, could result in data loss or damage to files. As such, it's recommended that you perform these exercises on an isolated network, using OS X computers and iOS devices that are not critical to your daily productivity. Apple, Inc., and Peachpit Press are not responsible for any data loss or any damage to any equipment that occurs as a direct or indirect result of following the procedures described in this guide.

Mandatory Requirements

Here's what you will need to complete the lessons in the guide:

- Two Mac computers, each with OS X Yosemite. One Mac is referred to as your "administrator computer," and the Mac on which you will install OS X Server is referred to as your "server computer" or, more simply, your "server." After you are done using your server computer with this guide, you should erase and reinstall OS X on its startup volume before using it again in a production environment.
- An Apple ID that is associated with a verified email address so you can obtain Apple Push Notification service (APNs) certificates for Server app notifications and for the Profile Manager service. You can create an Apple ID at the appropriate time during an exercise if you don't already have an Apple ID.
- A valid licensed copy of OS X Server from the Mac App Store.
- An Internet connection for obtaining APNs certificates for alerts and for the Profile Manager service.
- An isolated network or subnet with an exercise-specific configuration. This can be facilitated with something as simple as a small network Wi-Fi router with multiple Ethernet ports. For example, Apple AirPort Extreme would be a good choice. You can find instructions for the general setup of an exercise network and specific instructions for the configuration of AirPort Extreme at www.apple.com/airport-extreme.
- A router (such as AirPort Extreme) to connect the small isolated network to the Internet. It will be helpful to be familiar with how to configure it.
- Two Ethernet network cables (to complete the NetInstall exercises); each Ethernet cable will connect a Mac to the Ethernet switch.
- Student Materials demonstration files, which you can download after registering your guide with Peachpit. Instructions for registration and download are included in "Exer-

cise 1.1 Configure OS X Before Installing OS X Server on Your Server Computer ” on page 21.

Optional Add-Ons

If a specific resource is required for an optional exercise, it will be listed as a prerequisite at the beginning of that exercise. Here are some examples:

- An iOS device to test access to OS X Server services, including the Profile Manager service
- A Wi-Fi access point (preferably the same AirPort base station) to provide wireless access for iOS devices to your private network
- For “Exercise 19.1 Configure the DHCP Service (Optional)” on page 544: To provide DHCP on an extra isolated network: either an additional built-in Ethernet port on your Mac (for example, if your server computer is a Mac Pro) or a USB to Ethernet adapter or a Thunderbolt to Gigabit Ethernet Adapter; and an extra Ethernet network switch

If you lack the equipment necessary to complete a given exercise, you are still encouraged to read the step-by-step instructions and examine the screenshots to understand the procedures demonstrated.

Network Infrastructure

As was previously stated, the exercises require an isolated network. You should replicate the instructor-led classroom environment, which is described in the next sections, as closely as possible so that you do not need to translate between the exercise instructions and your situation.

IPv4 Addresses

The instructor-led environment provides an IPv4 network with a gateway of 10.0.0.1 and subnet mask of 255.255.255.0; if possible, configure your internal network with the same parameters.

Many consumer-level routers are configured with a gateway of 192.168.1.1 and a subnet mask of 255.255.255.0. You might not be able to change this on your router; in many cases you will be able to replace the “10.0.0” portion of an IPv4 address in the exercise with a value appropriate for your isolated network (for example, 192.168.1.171 instead of

10.0.0.171 for a server address for student 17). You will need to remember to substitute your network prefix throughout the exercises.

DHCP

The classroom DHCP service provides IPv4 addresses in the range of 10.0.0.180 to 10.0.0.254; if possible, configure your internal network's DHCP service with the same parameters. It will be helpful to know how to define the IP addresses of DNS servers being provided by DHCP.

If you can configure your isolated network's DHCP service, configure it to use a similar range of IPv4 addresses. If you are unable to change the range of IPv4 addresses, there is a possibility that the DHCP service will assign to a device an IPv4 address already in use by your server computer or your administrator computer. This is another reason to keep your network isolated; do not introduce new devices to it.

Domain Names

The exercises and reference material in this guide use the Internet domains pretendco.com, pretendco.private, and megaglobalcorp.com, which are for learning environments only; do not attempt to use these in your production environment.

The exercises are written in such a way that any existing DNS service on your isolated network will be ignored so that you can experience your server setting up the DNS service for itself.

Advanced Administrators

If you already have advanced server administration skills, you may choose to use different settings, including your organization's Internet domain (instead of pretendco.com), your organization's DNS service, and a different IPv4 address scheme, but be warned that this introduces a high level of variability that the exercises cannot address in the given space, and be prepared to modify the exercises on your own as necessary.

Exercise Order

The exercises in this guide are designed to be relatively independent of each other so that you can perform them out of order or skip exercises you are not interested in. However, some exercises you must perform in the correct order, and where appropriate, an exercise lists these prerequisites. Here are some examples:

- You must perform all the exercises in Lesson 1 "Installing OS X Server" to install OS X Server and configure your administrator computer before performing any other exercises.
- You must perform "Exercise 4.2 Configure an Open Directory Certificate Authority" on page 152 and "Exercise 9.1 Create and Import Network Accounts" on page 292 to create users who you will use in later exercises; otherwise, if the prerequisites for an exercise include the user account used in the lesson, you can simply create those user (and possibly) group accounts with the Server app's Users pane.

Apple Training and Certification

The Apple Training and Certification program is designed to keep you at the forefront of Apple technology. Certification creates a benchmark to demonstrate your proficiency in specific Apple technologies and can give you a competitive edge in today's evolving job market.

Certification exams are delivered at Apple Authorized Training Centers around the world.

Reading this guide or attending the Yosemite 201 class will help prepare you to pass the OS X Server Essentials 10.10 exam and become an Apple Certified Technical Coordinator. Passing both this exam and the OS X Support Essentials 10.10 exam earns Apple Certified Technical Coordinator (ACTC) 10.10 certification. This is the second level of the Apple certification program for Mac professionals, which includes the following:

- Apple Certified Support Professional (ACSP) certification verifies an understanding of OS X core functionality and an ability to configure key services, perform basic troubleshooting, and support multiple users with essential OS X capabilities. ACSP certification is designed for the help desk professional, technical coordinator, or power user who supports OS X users, manages networks, or provides technical support for the Mac. Students earn ACSP certification by passing the OS X Support Essentials 10.10 exam. Visit http://training.apple.com/certification/osxyosemite to review the OS X Support Essentials Exam Prep Guide. To prepare for this exam, attend the Yosemite 101 class or read *Apple Pro Training Series: OS X Support Essentials 10.10*.
- Apple Certified Technical Coordinator (ACTC) certification verifies a foundation in OS X and OS X Server core functionality and an ability to configure key services and perform basic troubleshooting. ACTC certification is intended for OS X technical coordinators and entry-level system administrators who maintain small to medium-

size networks of computers using OS X Server. Students earn ACTC certification by passing the OS X Support Essentials 10.10 exam and OS X Server Essentials 10.10 exam. Visit http://training.apple.com/certification/osxyosemite review the OS X Server Essentials Exam Prep Guide.

MORE INFO ► To read OS X technical white papers and learn more about all Apple certifications, visit http://training.apple.com.

NOTE ► Although all the questions in the OS X Server Essentials 10.10 exam are based on material in this guide, nothing can substitute for time spent learning the technology. After you read the guide or take the class, spend time increasing your familiarity with OS X Server on your own to ensure your success on the certification exam.

This page intentionally left blank

Lesson 4

Configuring SSL Certificates

You can use OS X Server without doing any additional work to secure its services. However, you can use the Secure Sockets Layer (SSL) technology to prove your server's identity to client computers and devices and to encrypt communication between your server and client computers and devices. This lesson starts by describing the basics of SSL and then shows you how to configure SSL certificates for use with OS X Server.

GOALS

- Describe the basics of SSL certificates
- Create a certificate signing request
- Create a self-signed SSL certificate
- Import a certificate signed by a certificate authority
- Archive your certificate
- Renew your certificate
- Configure which certificate your OS X Server services use

Reference 4.1
Describe SSL Certificate Basics

You want the users who use your server's services to trust your server's identity and to be able to encrypt network traffic with your server.

The OS X solution is to use SSL, which is a system for transmitting data securely between hosts. You can configure your server to use an SSL certificate, which provides the ability to use the SSL system.

An "SSL certificate" (also referred to as simply a "certificate") is a file that identifies the certificate holder. A certificate specifies the permitted use of the certificate and has an expiration date. Importantly, a certificate includes a public key infrastructure (PKI) public key.

PKI involves the use of public and private keys. Grossly simplified, a "key" is a cryptographic blob of data, and within PKI, public and private keys are created in a way that they are mathematically linked: Data encrypted with one key can be decrypted only by using the other key. If you can decrypt data with one key, it proves that the data was encrypted with the other key. The public key is

made publicly available, and the private key should be kept private. Fortunately, all of this encryption and decryption happens behind the scenes and is the basis for establishing secure communications.

Here are some definitions:

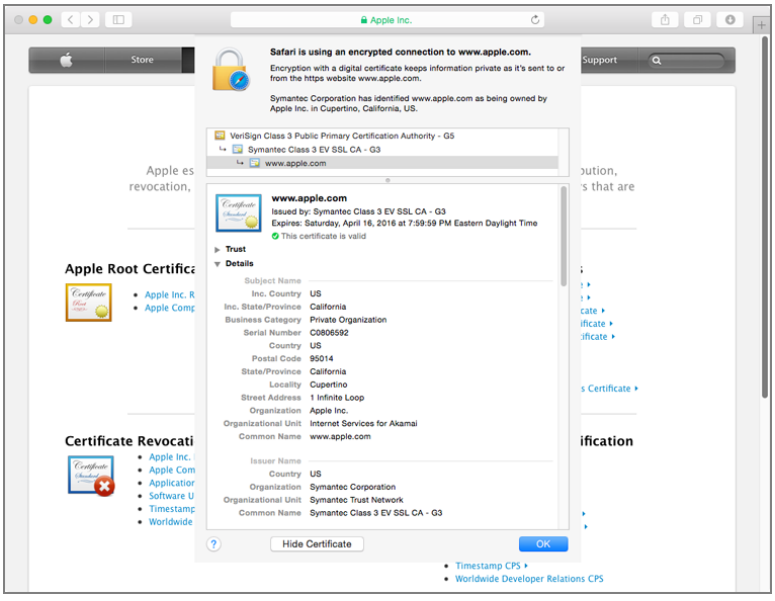
A “digital identity” (or more simply, an “identity”) is an electronic means of identifying an entity (such as a person or a server). An identity is the combination of a certificate (which includes the public key) and the corresponding private key. If you don’t have your private key, you can’t prove your identity. Similarly, if another entity has your private key, that other entity can claim your identity, so be sure to keep your private key private!

Again simplifying, a “digital signature” is a cryptographic scheme that uses PKI private and public keys to demonstrate that a given message (a digital file such as an SSL certificate) has not been changed since the signature was generated. If a message, which has been signed, changes or is otherwise tampered with, it will be clear that the signature no longer matches the underlying data. Therefore, you can use a digital signature on a certificate to prove its integrity.

A certificate must be either self-signed or signed by a “certification authority” (also known as a “certificate authority" or, more simply, a “CA”). In other words, you can sign your own certificate using your private key (remember that a certificate is a file that identifies the holder of the certificate and includes the public key), or you can have someone else, namely, a CA, use their private key to sign your certificate.

An “intermediate CA" is a CA whose certificate is signed by another CA. So, it’s possible to have a hierarchical “chain” of certificates, where an intermediate CA, which in turn is signed by yet another CA, signs a certificate.

In the following figure, the certificate for www.apple.com is signed by an intermediate CA with the name of Symantec Class 3 EV SSL CA - G3, and that intermediate CA is signed by a CA with the name of VeriSign Class 3 Public Primary Certification Authority - G5.

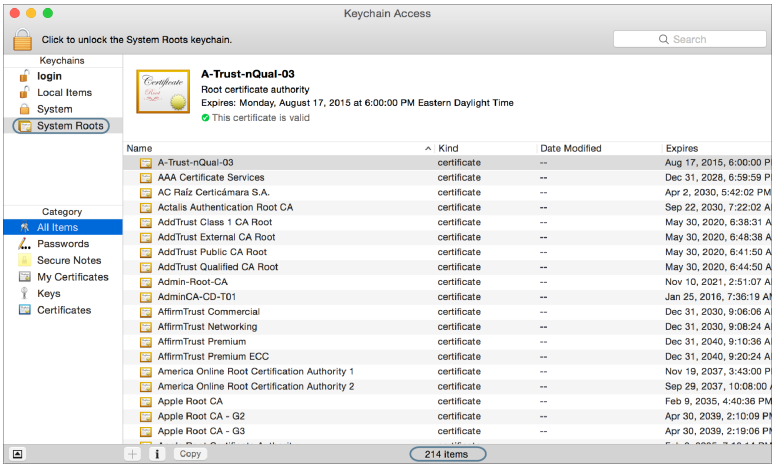


You can follow a chain of certificates, starting with a signed certificate, up to the intermediate CA and ending at the top of the chain. The certificate chain ends with a CA that signs its own certificate, which is called a "root CA." It is not required to have an intermediate CA involved—you could simply have a root CA sign your certificate—but in practice, an intermediate CA is often involved.

How do you know if you can trust a CA? After all, since a root CA has signed its own SSL certificate, this effectively means that the organization in control of a root CA simply asserts that you should trust that it is who it claims to be.

The answer is that trust has to start somewhere. In OS X and iOS, Apple includes a collection of root and intermediate CAs that Apple has determined are worthy of trust (see the Apple Root Certificate Program page on the Apple site for the acceptance process: www.apple.com/certificateauthority/ca_program.html). Out of the box, your Mac computers and iOS devices are configured to trust those CAs. By extension, your Mac computers and iOS devices also trust any certificate or intermediate CA whose certificate chain ends with one of these CAs. In OS X, these trusted CAs are stored in the System Roots keychain. (See Lesson 8, "Keychain Management," in *Apple Pro Training Series: OS X Support Essentials 10.10* for more information about the various keychains in OS X.) You can use Keychain Access to view this collection of trusted root CAs. Open Keychain Access (in the Utilities folder). In the upper-left Keychains column, click System Roots. Note that in the

following figure the bottom of the window states that there are more than 200 trusted CAs or intermediate CAs by default in Yosemite.



MORE INFO ▶ Some third-party software companies, such as Mozilla, do not use the System Roots keychain and have their own mechanism to store CAs that their software is configured to trust.

In Lesson 8 "Configuring Open Directory Services", you will learn that when you configure your server as an Open Directory master, the Server app automatically creates a new CA and a new intermediate CA and uses the intermediate CA to sign a new SSL certificate with your server's host name as the common name (the Common Name value is part of what identifies the certificate holder). It is recommended that if you haven't engaged a widely trusted CA to sign an SSL certificate for your server, you should use the SSL certificate signed by your Open Directory intermediate CA; in Lesson 10 "Configuring OS X Server to Provide Device Management", you will learn how to use the Trust Profile to configure your iOS devices and OS X computers to trust your Open Directory CA and, by extension, the intermediate CA and the new SSL certificate.

But what about computers and devices that are outside your control and that you cannot configure? When people use computers and devices that are not configured to trust your server's self-signed SSL certificate or your server's Open Directory CA or intermediate CA and they try to securely access services on your server, they will still see a message that the identity of your server cannot be verified.

One way to prove your identity is for your server to use an SSL certificate that's signed by a CA that most computers and devices are configured to trust or trust inherently.

Deciding What Kind of Certificate to Use

Before going through the process of getting a widely trusted CA to sign a certificate for you, consider the services you'll use with the certificate, as well as the computers and devices that will access those services.

If you use a self-signed certificate, there is no additional server configuration to install the certificate on your server, but you do need to configure each client to trust that self-signed certificate. For a Mac client, this involves not only distributing the certificate to the Mac and adding it to the System keychain but also configuring how the operating system (OS) will trust the certificate.

> **NOTE ▶** If you use a self-signed certificate and are not able to configure all devices to trust that self-signed certificate, when users encounter a service that uses the self-signed certificate, they will be presented with a dialog informing them that the certificate may not be trustworthy and that to access services they must click Continue. This may undermine your efforts to train users not to automatically trust untrusted, expired, or otherwise invalid certificates.

If you use a certificate signed by a widely trusted CA, you need to generate a certificate signing request (CSR), submit the CSR to a CA, and then import the signed certificate.

Of course, it is possible to use a mix of certificates for different services; if your Websites service responds to multiple host names, you'd want a certificate for each host name that you use for web services secured by SSL.

In all cases, you need to configure your server's services to use the appropriate certificates.

The next section shows you how to obtain a certificate that's signed by a widely trusted CA so that you can use it to prove the identity of your server and to encrypt communications between your server and the users of your server's services.

Reference 4.2
Configuring SSL Certificates

Your server has a default SSL certificate that's self-signed. That's a good start, but no other computers or devices will trust services that use that certificate without additional configuration. To get a CA to sign a certificate, start by using the Server app to create a certificate signing request. Specific steps to accomplish this objective follow in more detail, but generally they include the following:

- Generating a new CSR
- Submitting your CSR to a CA that is generally trusted
- Importing the signed certificate
- Configuring your server's services to use your newly signed certificate

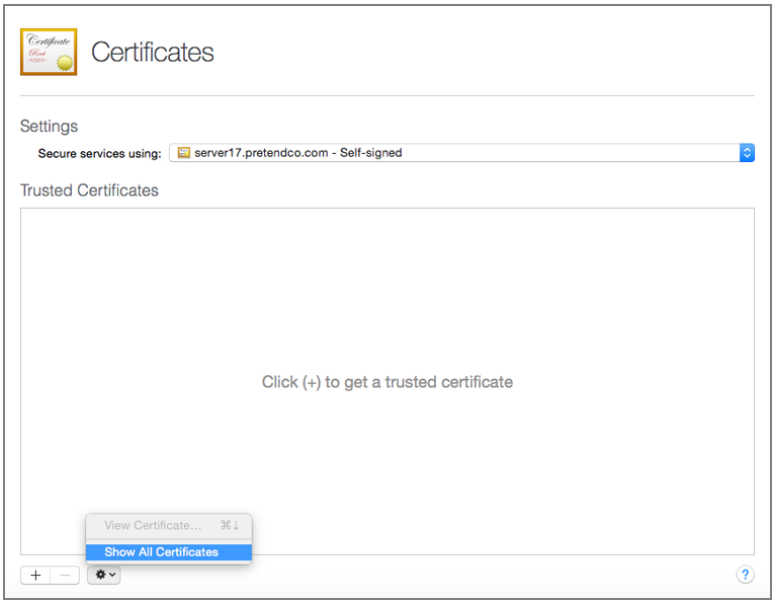
The CA's process of using your CSR and signing your SSL certificate with its own private key includes verifying your identity (otherwise, why would anyone trust the CA if it signed certificates from unverified entities?) and optionally charging you money.

To finish the story, computers and devices can now use your server's services without getting a warning that your SSL certificate is not verified (as long as those computers and devices trust the CA you've chosen to sign your certificate). Additionally, your server and the users of its services can use your server's SSL certificate in the process of encrypting communications for services that use that SSL certificate.

Before you start creating new certificates, take a moment to inspect what you already have.

Viewing Your Server's Default Certificate

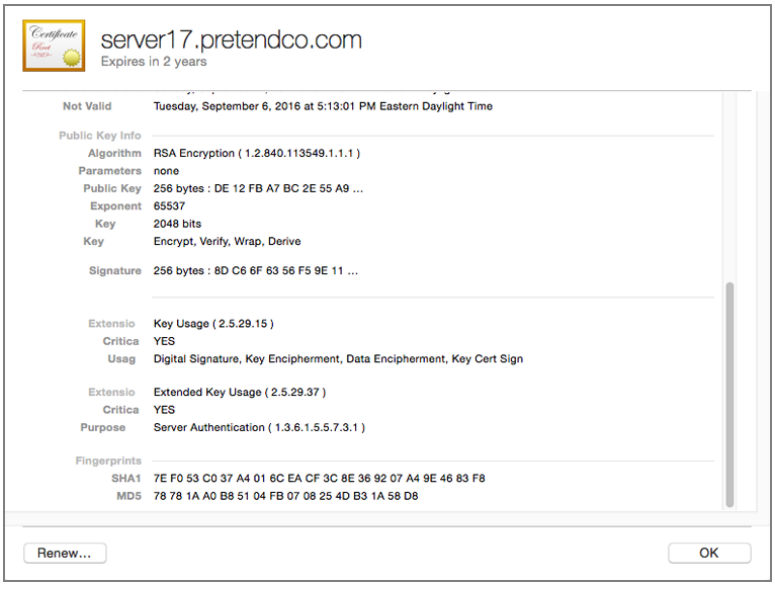
You can use the Server app to display certificates (if you're logged in at the server, you can also use the Keychain Access app). By default, the Server app doesn't display the default certificate. To display all certificates, select Certificates in the Server app sidebar, and then from the Action (gear icon) pop-up menu, choose Show All Certificates.



Once Show All Certificates is selected, you see your default certificate. In the following figure, the certificate has the server's host name and expires in two years.

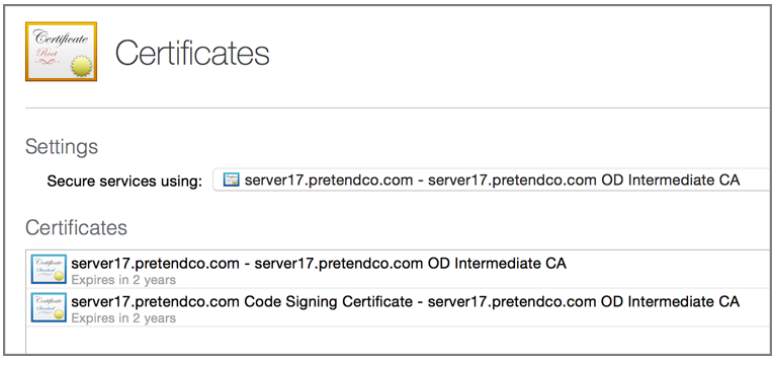
MORE INFO ▶ When you use the Server app Change Host Name Assistant to change your server's host name, it automatically creates a new self-signed certificate for the new host name.

To get more details, double-click the certificate; alternatively, select it, click the Action (gear icon) pop-up menu, and choose View Certificate. When you choose View Certificate, the details of the certificate appear. You'll need to scroll to inspect all of the certificate's information.



Click OK to return to the Certificates pane.

The following figure illustrates what you'd see after you configure your server as an Open Directory master or replica. At first glance, it looks like there is just one additional certificate, the code signing certificate, but the certificate with the server's host name is no longer a self-signed certificate but a certificate signed by your Open Directory CA; that certificate icon is blue, whereas the original self-signed certificate was bronze.



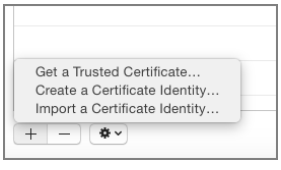
Explaining Options for Adding New Certificates

In the Server app Certificates pane, if Show All Certificates is not selected in the Action (gear icon) pop-up menu, then the text in the Trusted Certificates field instructs you to "Click (+) to get a trusted certificate." This is the path to getting a CA to sign a certificate for you.

> **NOTE ▶** To reveal the menu choice to create a new self-signed certificate, you must first click the Action (gear icon) pop-up menu and choose Show All Certificates.

After Show All Certificates has been selected, click Add (+) to reveal three menu commands:

- Get a Trusted Certificate has the same effect as clicking Add (+) if Show All Certificates is not selected; it allows you to quickly generate a certificate signing request.
- Create a Certificate Identity is the command to choose to create a new self-signed certificate.
- Import a Certificate Identity allows you to import a signed certificate or a certificate and private key that you've archived.



Obtaining a Trusted Certificate

You can choose to get a CA to sign a certificate for you so that users around the world can use your server's services without being notified that your server's identity is not verified.

> **NOTE ▶** In versions of OS X Server previous to 2.2, you need to first create a self-signed certificate and then create a CSR from that certificate. This process has been streamlined as of OS X Server version 2.2. However, be aware that when you use the following procedure to generate a CSR, the Server app generates a public and private key pair, but it doesn't generate a self-signed certificate.

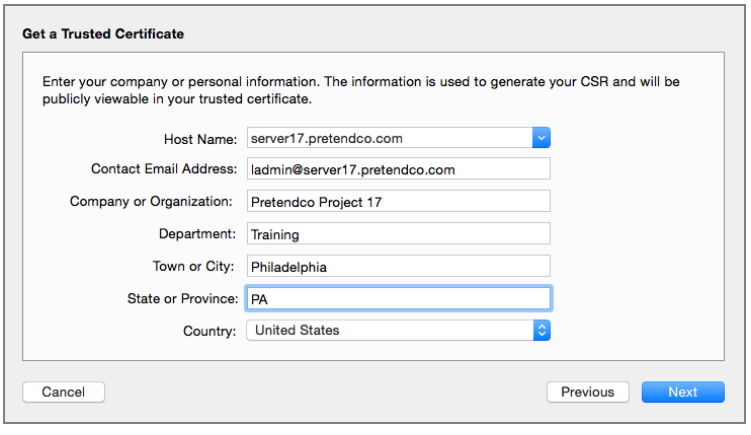
The path to generating a self-signed certificate depends on whether Show All Certificates is selected in the Action (gear icon) pop-up menu:

- If Show All Certificates is not selected, simply click Add (+) in the Certificates pane.
- If Show All Certificates is selected, click Add (+), and then choose Get a Trusted Certificate.

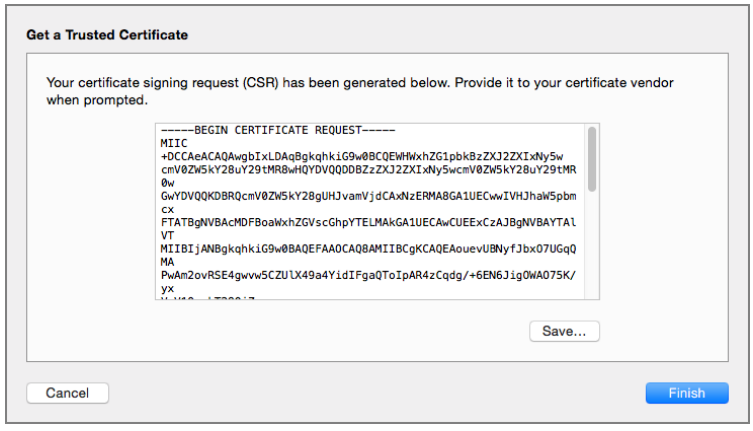
After that, you'll see the Get a Trusted Certificate wizard.

In the next pane you can enter all the information that is necessary to establish an identity. A CA uses these details to verify your identity.

In the Host Name field, enter the host name you'll use for the services that will use this certificate. Use your organization's full legal name for the Company or Organization field, or if it's for personal use, just use your full name. The Department field is flexible; you can enter information such as your department name, but you should enter some value. To be fully compliant with standards, do not abbreviate your state or province. The following figure illustrates all of the fields completed.

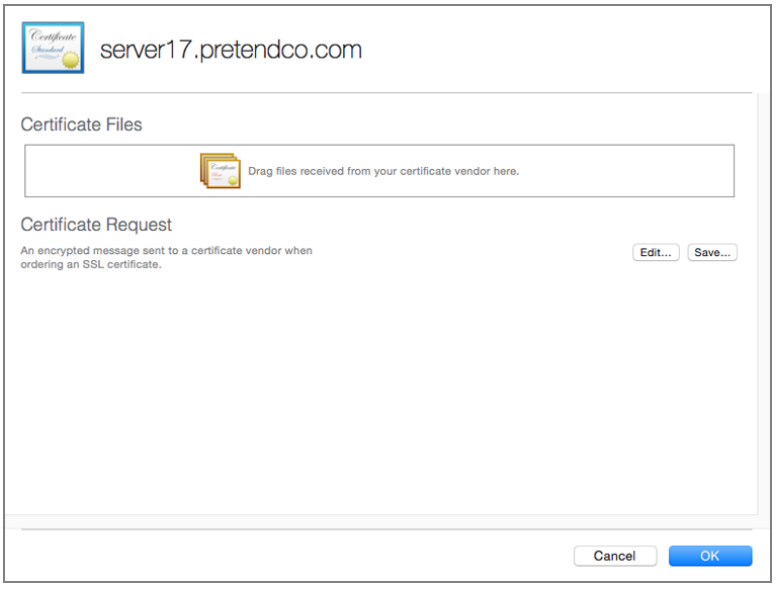


The next pane displays the text of your CSR, which you will submit to the CA of your choice. You can wait and access this text later, or you can select and copy this text, or click Save, now.

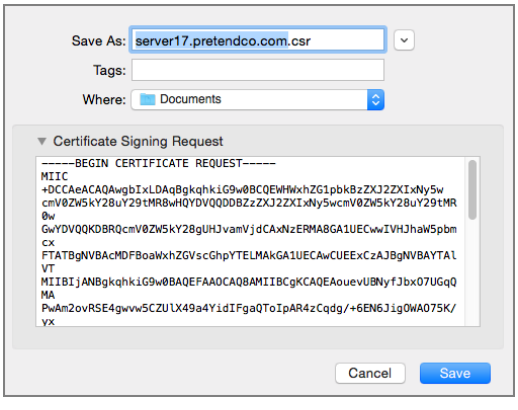


After you click Finish, the Server app displays the pending request.

If you didn't copy the text of your CSR earlier, you can access it again: Select the certificate that's marked "Pending," click the Action (gear icon) pop-up menu, and choose View Certificate Signing Request (or just double-click the pending certificate item).



Your course of action depends on how your CA accepts CSRs. If your CA allows you to upload a text file, then use the Save dialog to save the CSR as a text file. If your CA requires you to paste the text of the CA into a web form, click the disclosure triangle, and then copy the text of the CSR.



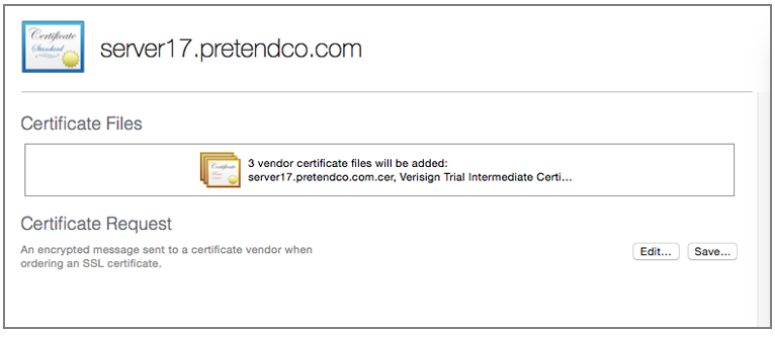
You need to choose an appropriate CA for your organization's needs (choosing a CA is outside the scope of this guide), send the CSR to the CA, and prove your identity to the CA. After some period of time, you will receive a signed certificate from the CA.

Importing a Signed Certificate

Once you receive the signed certificate from the CA, it's time to import it with the Server app. If you are still at the list of certificates, double-click your pending certificate to reveal the field into which you can drag your signed certificate.

> **NOTE ▶** If the CA provides you with the certificate in text form rather than in a separate file, you'll need to convert that text into a file. A quick way to do this is to select and copy the text, open TextEdit, press Command-N to create a new file, and choose Format > Make Plain Text (if that is an available command). Paste the text into the text file, and save it with a .cer extension.

Double-click the pending CSR, and drag the file containing a signed certificate, as well as any ancillary files provided by the CA, into the Certificate Files field (this is also where you could import a certificate and private key that you've exported with Keychain Access). Once the certificate is in the Certificate Files field, its color will be blue, as long as the top of the certificate chain is a root CA that your server trusts.



> **NOTE ▶** If you click Edit next to Certificate Request and then click Edit in the confirming dialog, a new public and private key pair and a new CSR will be generated, and you'll lose the original CSR.

Click OK to save your changes.

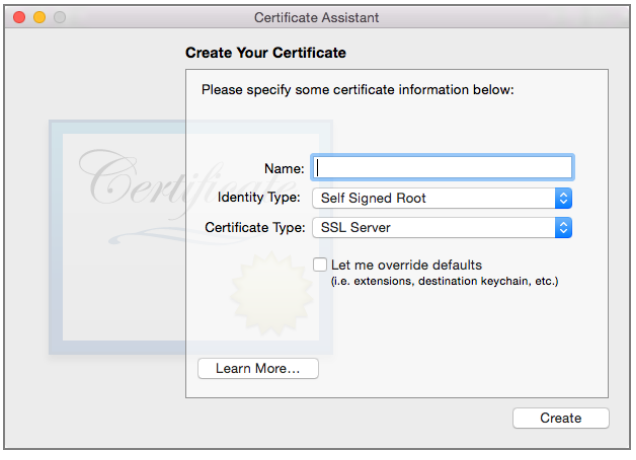
Generating a Self-Signed Certificate

In addition to generating a CSR, you can also use the Server app to generate a new self-signed certificate. This is useful if your server offers services at an alternative host name that corresponds to your server's Internet Protocol version 4 (IPv4) address or another IPv4 address that your server is configured to use and if you have the ability to configure computers and iOS devices to trust the self-signed certificate.

MORE INFO ▶ In versions of the Server app prior to version 2.2, the workflow was to create a self-signed certificate, then generate a CSR, and finally replace the self-signed certificate with the signed certificate. In version 2.2 and later, the Server app does not offer a way to replace a self-signed certificate with a signed certificate.

In the Certificates pane, when you click Add (+) and choose Create a Certificate Identity, you see a blank Name field.

NOTE ▶ The Show All Certificates option must be selected in the Action (gear icon) pop-up menu for the Create a Certificate Identity command to be available from the Add (+) button.



Enter the host name for the self-signed certificate, and then click Create.

NOTE ▶ You can select the "Let me override defaults" checkbox if you have more specific needs, but for most purposes, the defaults will suffice.

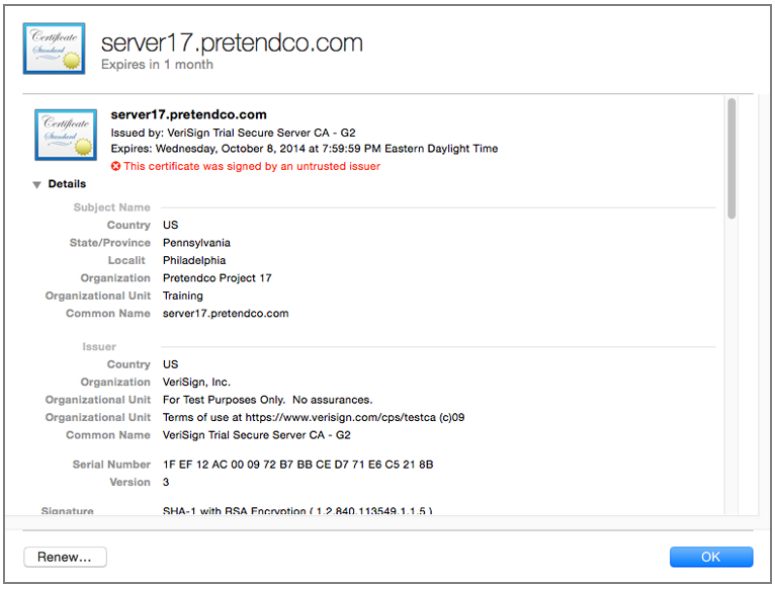
At the warning that you are about to create a self-signed certificate, click Continue.

At the Conclusion window, click Done. Finally, click either Always Allow or Allow to allow the Server app to copy the public and private key pair and the certificate from your login keychain to the System keychain and to /private/etc/certificates/.

You'll see the certificate in the Certificates field, as long as the Show All Certificates option is selected in the Action (gear icon) pop-up menu.

Inspecting a Certificate

You can inspect your certificates with the Server app, as well as with the System keychain of your server computer (the System keychain contains items that are not user specific and that are available to all users of a system). The following figure shows a certificate that's been signed by a CA for test purposes. Note that the OS has not yet been configured to trust the CA that signed this certificate.



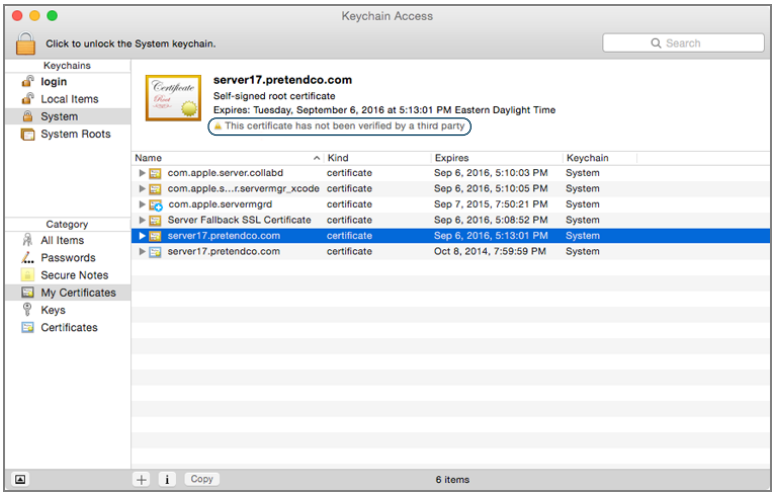
You can also use Keychain Access to inspect a certificate and its associated private key. Because the certificate and private key are stored in the System keychain on the server, you need to log in directly on your server (or use a screen-sharing method to control your server) to use Keychain Access to access the private key.

Keychain Access is in the /Applications/Utilities/ folder on your startup volume; you can use Spotlight or Launchpad to search for it (in Launchpad, it is in the folder named Other). Select the My Certificates category to filter the items that Keychain Access displays. If necessary, toggle the show/hide button in the lower-left corner of the Keychain Access window until you can see all keychains. Select the System keychain to show items that are for the entire system, not just for the user who is currently logged in.

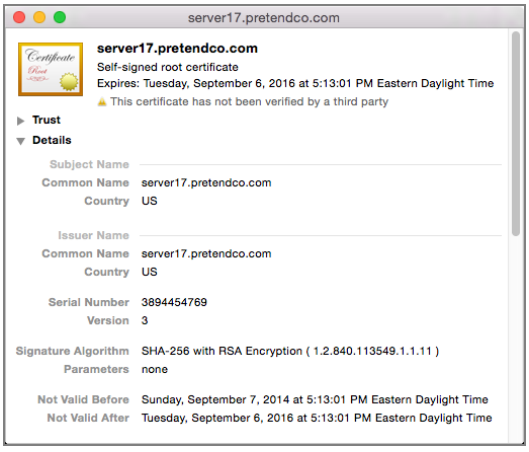
At least three items are listed (if you provided an Apple ID for push notifications, you will see more items):

- com.apple.servermgrd, which is used for remote administration with the Server app
- A certificate named Server Fallback SSL Certificate, which the Server app automatically uses if the default SSL certificate is removed
- An SSL certificate with the host name of your server

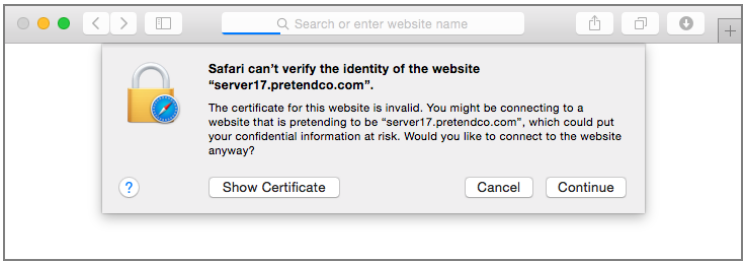
When you select a certificate that is not signed by a trusted CA, Keychain Access displays a warning icon, along with the text that explains the issue. In the following figure, the warning for the self-signed certificate is "This certificate has not been verified by a third party."



If you double-click your default self-signed SSL certificate to open it, you'll see a warning icon and the text "This certificate has not been verified by a third party."



If a service on your server uses this self-signed certificate, when users attempt to use services that use that SSL certificate, they may be warned that your SSL certificate is not trusted, as shown in the following figure.



It's recommended to train your users that when they see an SSL warning, they should *not* continue using the service that uses the unverified SSL certificate.

Archiving Your Certificate

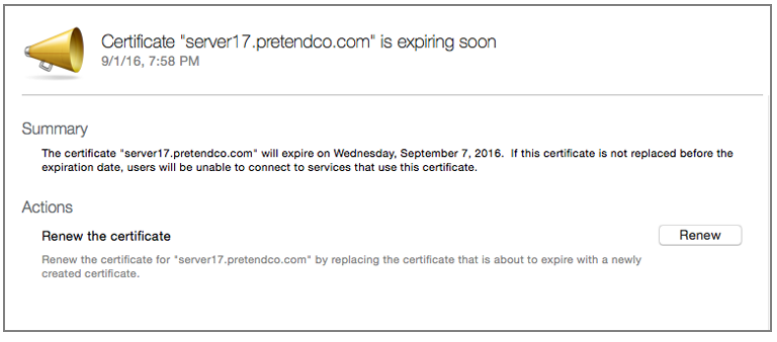
Whether you have a self-signed certificate or a certificate signed by a CA, you should take steps to archive your certificate and its private key. You may need to reinstall your server in the future, or an administrator might accidentally remove your certificate and its private key; if you have an archive of your certificate and private key, you can easily use the Server app to re-import your certificate and its private key.

You use the Keychain Access app to export your certificate and private key. Keychain Access prompts you to specify a password to protect your private key; it is recommended that you use a strong password.

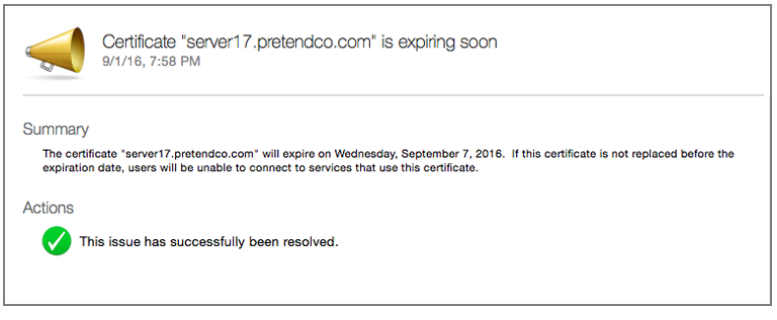
You use the Server app to import the certificate and private key. You need to provide the password that was entered when the certificate was exported in the first place; otherwise, you will not be able to import.

Renewing Your Certificate

SSL certificates do not last forever. Luckily, it is simple to renew SSL certificates. The Server app issues an alert when an SSL certificate expiration date approaches. To renew a self-signed SSL certificate, simply click Renew when viewing the certificate in the Certificates pane or when viewing the alert.

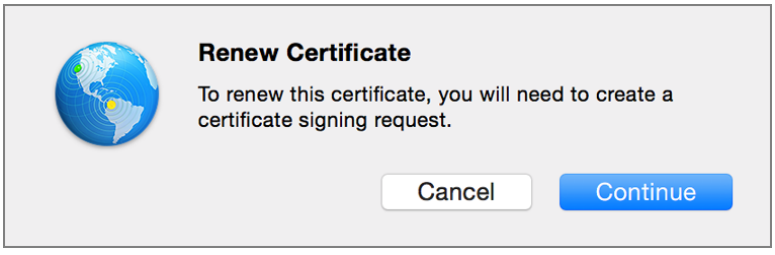


Once you click Renew, the Server app takes care of renewing the certificate, and the alert displays that the issue has been resolved.



NOTE ▶ Do not click Renew for an Open Directory CA because this causes changes to the CA properties, and your Open Directory intermediate CA will no longer be signed by a trusted authority.

If you have a certificate signed by a widely trusted CA, when you click Renew, you will see the message that you need to generate a new CSR. See the earlier section "Obtaining a Trusted Certificate" on page 131 for more details.



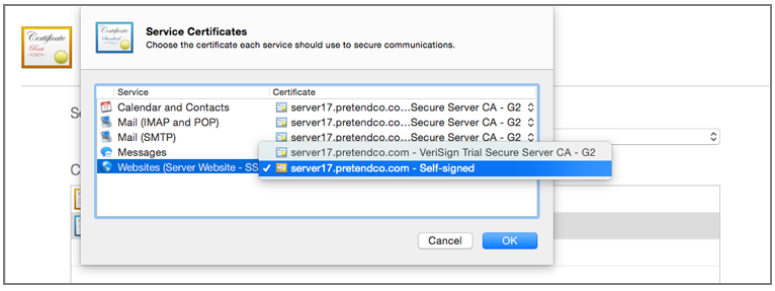
Configuring OS X Server Services to Use a Certificate

Once you have taken steps to obtain a signed certificate or create a new self-signed certificate or have configured your server as an Open Directory server, you should use the Server app to configure services to use that certificate. You start in the Certificates pane of the Server app.

With the pop-up menu, you can do either of the following:

- Choose one certificate to specify that all services use that certificate.
- Choose Custom to configure each service separately to use or not use a certificate.

The following figure shows an example of choosing Custom and then editing the value for the default secure site of the Websites service. Note that there are some extra certificates in the figure. This illustrates that you can configure your server to respond to requests at multiple host names, create a certificate for each host name, and configure each secure site to use the appropriate certificate.



You can use the Server app to configure the following OS X Server services to use SSL:

- Calendar and Contacts
- Mail (IMAP and POP)
- Mail (SMTP)
- Messages
- Open Directory (appears only after starting Open Directory services)
- Websites

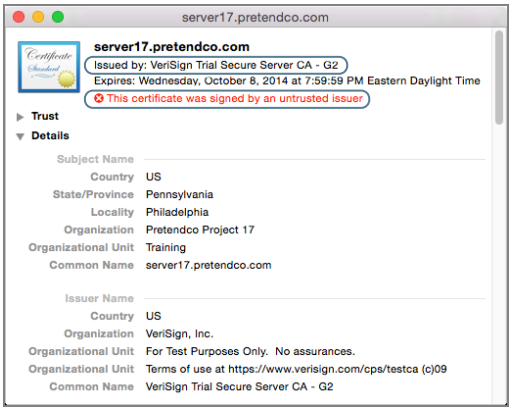
You will see in Lesson 20 "Hosting Websites" that you can granularly specify an SSL certificate for each website you host, and you can use the Profile Manager pane to specify the SSL certificate to use for the Profile Manager service.

A few other services use SSL but do not appear in the Server app:

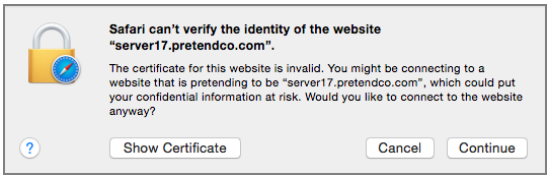
- com.apple.servermgrd (for remote administration with the Server app)
- VPN
- Xcode

Following the Certificate Chain

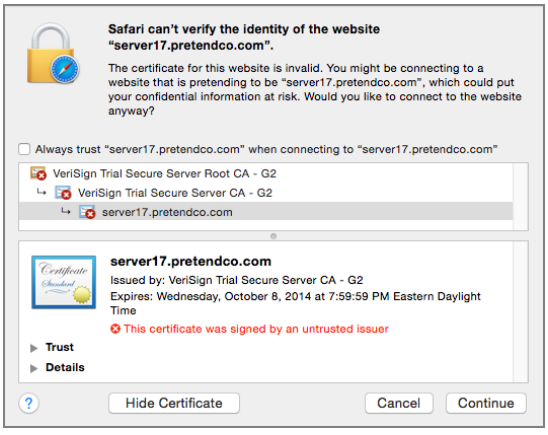
When choosing a CA to use, make sure that it's a root CA that most computers and devices are configured to trust. It's not useful for you to have a CA sign your certificate if not many computers or devices will trust that certificate. As an example, the following figure shows how an SSL certificate signed by a trial CA appears in Keychain Access.



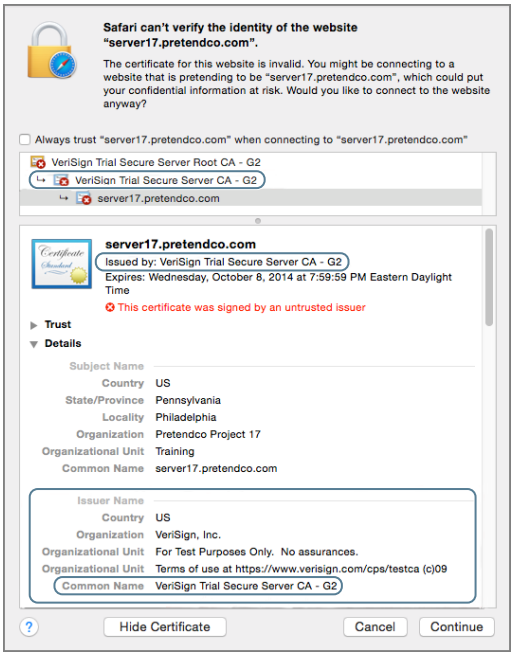
You can see that the "Issued by" field near the top of the window shows VeriSign Trial Secure Server CA – G2. Note the red X icon and the text "This certificate was signed by an untrusted issuer." This is a CA that is by default not trusted by computers and devices, so even if you used this signed certificate for OS X Server services, the people who access your services would experience trouble. In some cases, the service might silently fail, or the user may be alerted that the identity of the service cannot be verified. The following figure illustrates that on a client Mac Safari notifies the user that Safari can't verify the identity of the website.



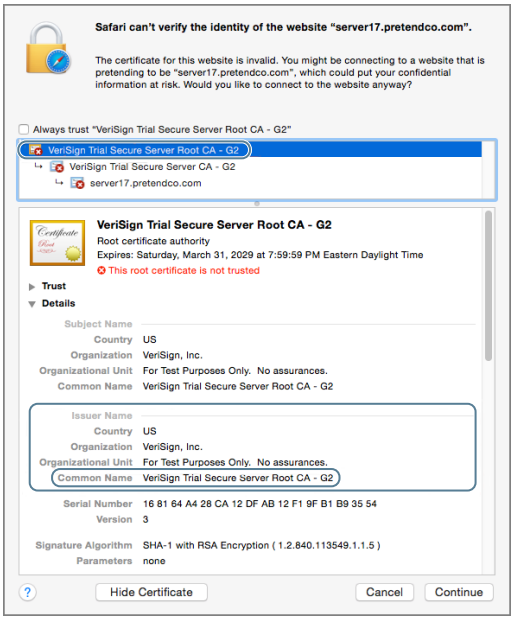
If you click Show Certificate, Safari displays the certificate chain. The following figure shows what you see when you select the server's certificate at the bottom of the certificate chain: that the certificate was signed by an untrusted issuer.



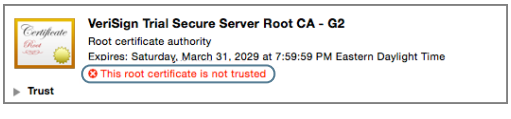
The following figure illustrates that if you click the Details disclosure triangle, you'll see information about the identity of the certificate holder, as well as information about the issuer (the entity that signed the certificate). In this case, the issuer's common name is VeriSign Trial Secure Server CA – G2.



When you select the certificate in the middle of the certificate chain, you see that this is an intermediate CA; the window states "Intermediate certificate authority," and the Issuer Name information shows you that the common name of the issuer (or signer) is VeriSign Trial Secure Server Root CA – G2.



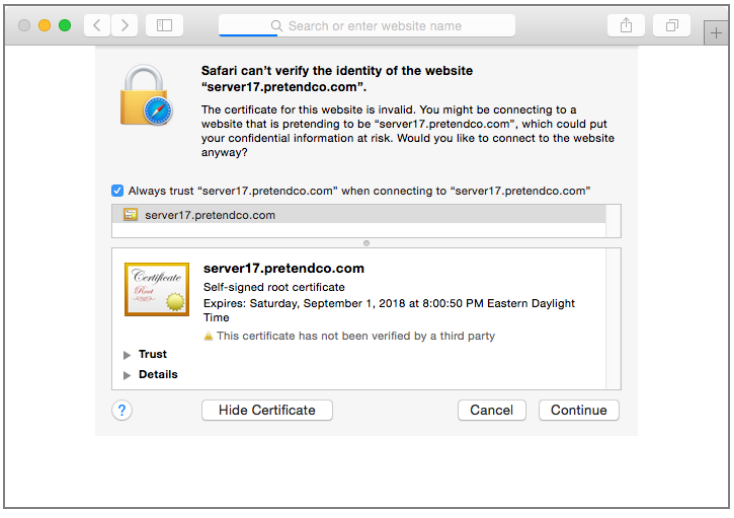
Finally, when you select the certificate at the top of the certificate chain, you see that this is a root CA; the window states "This root certificate is not trusted." This root CA is not in this computer's System Root keychain, so Safari doesn't trust the intermediate CA, and it doesn't trust the server17.pretendco.com certificate either.



Since that example root CA is for trial use only, it is not recommended to configure your Mac to always trust it outside of a learning or testing environment.

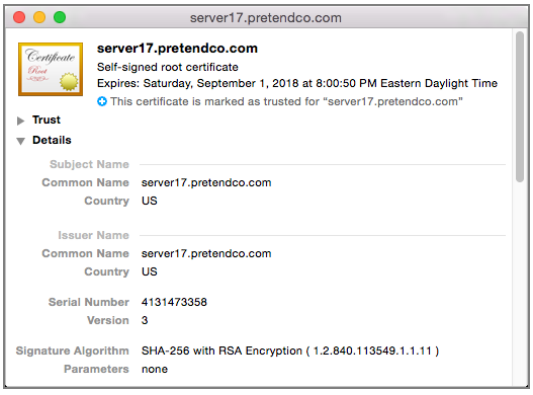
Configuring Trust

You can configure your Mac to always trust a certificate for the currently logged-in user. Returning to the previous example of your server using its self-signed SSL certificate for a website, you can click Show Certificate and then select the "Always trust..." checkbox.

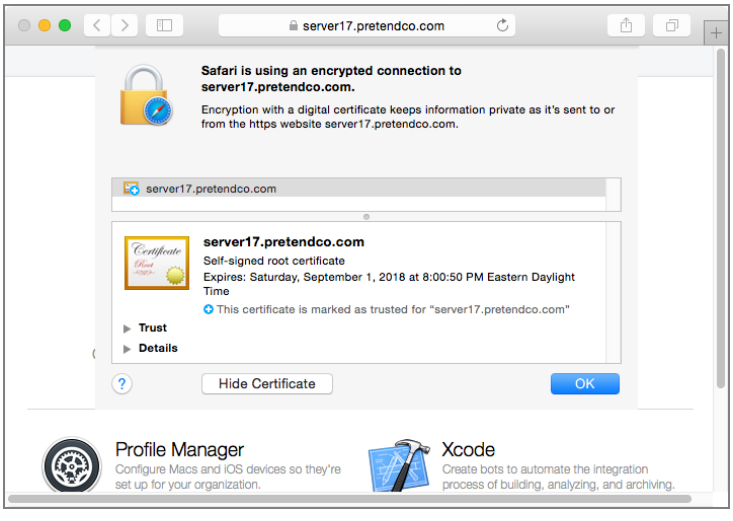


After you select "Always trust...," OS X asks for your login credentials. After you successfully authenticate, OS X adds the certificate to your personal login keychain and configures your system to always trust the certificate for SSL purposes so that your Mac trusts it when you are logged in with the account that you were logged in as when you clicked the "Always trust..." checkbox. This will not affect any other computers or devices or any other users who log in to that Mac.

In Keychain Access, you can open and inspect the self-signed certificate you just added. Note the blue plus (+) icon with the text that states the certificate is marked as trusted for server17.pretendco.com.

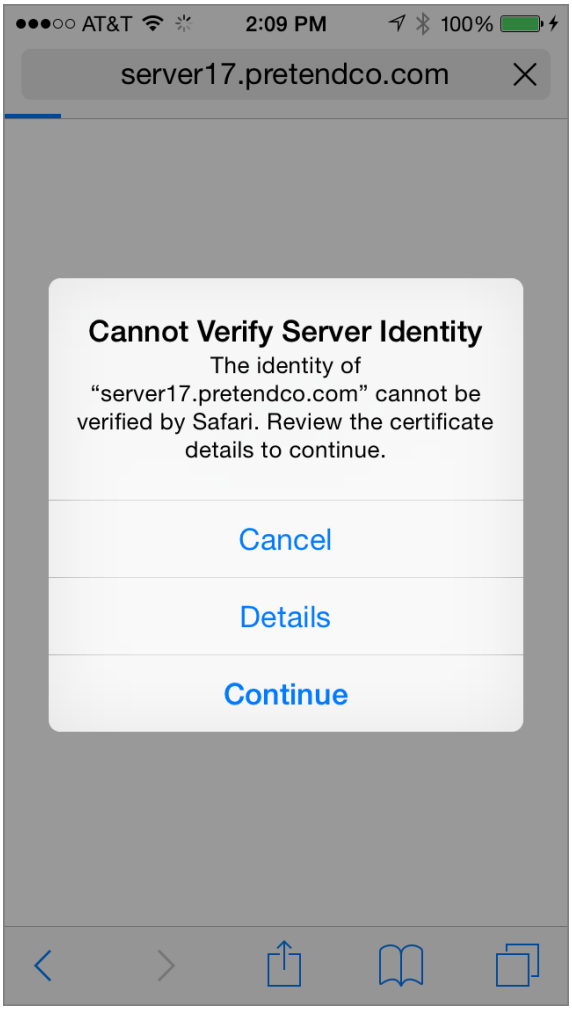


After you visit the site again in Safari, if you click the encryption icon in the Address and Search field and then click Show Certificate, you see similar information.

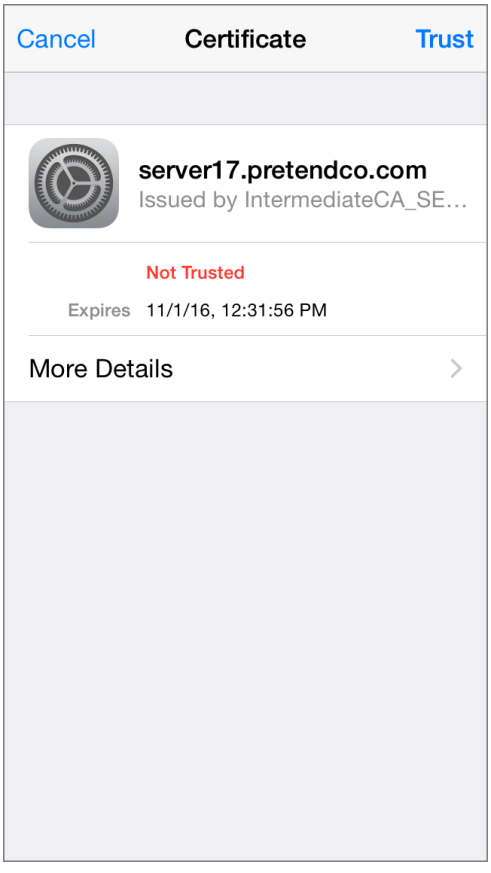


A further option for Mac computers is to download and install the certificate in the System keychain, with the "Always trust..." checkbox selected for SSL. Keep in mind that you would need to do this for *every* Mac that uses SSL-enabled services from your server.

For an iOS device, when you open Safari to a page protected by the server's self-signed certificate, you can tap Details.



Then tap Trust.



Now your iOS device is configured to trust that certificate.

Note that you can use a configuration profile to distribute a certificate to Mac computers and iOS devices. This automatically configures the device to trust the certificate. See Lesson 11 "Managing with Profile Manager" for more information about profiles.

See "Exercise 4.3 Configure Your Administrator Computer to Trust an SSL Certificate" on page 159 for complete instructions.

Reference 4.3 Troubleshooting

Certificate Assistant uses the IPv4 address of the Mac from which you run the Server app, so if you're using an administrator computer to configure a remote server and generate a

new self-signed certificate, be sure to use the server's host name and IP address where appropriate.

When you configure your server as an Open Directory server, if you have a self-signed certificate with your server's host name in the certificate's Common Name field, the Server app replaces the original self-signed SSL certificate with a new certificate. This new certificate will be signed by a newly created intermediate CA associated with your server's Open Directory service.

However, if you have a certificate with your server's host name in the certificate's Common Name field and the certificate is signed by a CA or an intermediate CA (that is not associated with your Open Directory service), then the Server app doesn't replace it with a new one signed by the Open Directory intermediate CA (however, the Server app still creates the Open Directory CA and intermediate CA).

Each certificate has an expiration date; if the current date is later than a certificate's expiration date, the certificate is not valid.

MORE INFO ▶ Some files associated with certificates are stored in /private/etc/certificates/, and possibly /private/var/root/Library/Application Support/Certificate Authority/, on your server.

Exercise 4.1
Examine the Default SSL Certificate

Prerequisites

- All exercises in Lesson 1 "Installing OS X Server"
- "Exercise 2.1 Create DNS Zones and Records" on page 71

In this exercise, you will examine the default self-signed certificate.

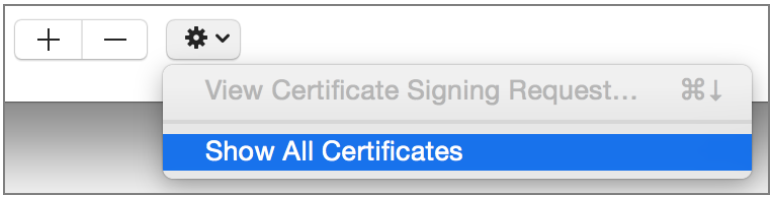
1 Perform these exercises on your administrator computer. If you do not already have a connection to your server computer with the Server app on your administrator computer, then connect to it with the following steps: Open the Server app on your administrator computer, choose Manage > Connect to Server, select your server, click Continue, provide administrator credentials (Administrator Name: **ladmin** Adminis-

trator Password: ladminpw), deselect the "Remember this password" checkbox, and then click Connect.

2 In the Server app sidebar, select Certificates.

3 Note that by default your server's services use a certificate that is self-signed.

4 Click the Action (gear icon) menu, and choose Show All Certificates.



5 Select the self-signed certificate.

6 View the details of the certificate. Double-click the self-signed certificate, or click the Action menu and choose View Certificate.

7 Click OK return to the Certificates pane.

Note that there is little identifying information associated with this certificate. For example, there is no email address, organization name, department, or city. By default, no other computer or device trusts this self-signed certificate. To use this self-signed certificate to secure your server's services, you could configure your client computers and devices to trust this certificate.

Alternatively, you could click the Add (+) button and choose Get a Trusted Certificate, send the resulting certificate signing request to a widely trusted certificate authority to sign, and then import the signed certificate. However, this is outside the scope of this guide, so the next exercise is a compromise between using a self-signed certificate with little information and using a certificate signed by a widely trusted CA.

Exercise 4.2
Configure an Open Directory Certificate Authority

When you configure your server as an Open Directory (OD) master, the Server app automatically creates an OD CA, an intermediate CA, a signed certificate, and a code signing certificate that you can use with the Profile Manager service. When you enroll your Mac

computer or your iOS device with your server's Profile Manager service, your computer automatically trusts your server's OD CA. Additionally, if you bind your Mac to your OD server, it automatically trusts your server's OD CA. This guide has not yet covered binding or enrolling, so in "Exercise 4.3 Configure Your Administrator Computer to Trust an SSL Certificate" on page 159 you will use Safari to configure your administrator computer to trust your server's OD CA.

In this exercise, you will configure your OD CA. You will examine the new CA, the intermediate CA, and two new certificates and verify that the Server app automatically removes your server's old default self-signed certificate, updates services to use the certificate signed by the intermediate CA, and configures your server to trust the new certificates.

Configure Open Directory

Because the Server app creates keychain entries on your server, perform the following steps on your server.

Correct DNS records are crucial to the proper functioning of Open Directory services, so double-check DNS before starting the Open Directory service.

1. On your administrator computer, quit the Server app if it is open.
2. On your server computer, open Network Utility (use Spotlight if necessary).
3. Click the Lookup tab.
4. Enter your server's host name in the field, and then click Lookup.
5. Confirm that your server's IPv4 address is returned.
6. Enter your server's primary IPv4 address in the field, and then click Lookup.
7. Confirm that your server's host name is returned.

Once you've confirmed your DNS records, configure your server as an Open Directory master.

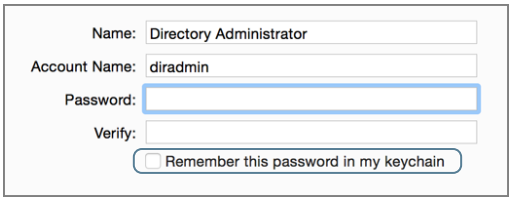
1. On your server, open the Server app, select your server, click Continue, provide administrator credentials (Administrator Name: **ladmin**, Administrator Password: **ladminpw**), deselect the "Remember this password" checkbox, and then click Connect.
2. If the Server app does not display the list of advanced services, hover the pointer above the word "Advanced" in the sidebar, and then click Show.

3 Select Open Directory.

4 Click On to turn on the Open Directory service (or in the Server app sidebar, Control-click Open Directory, and choose Start Open Directory Service).

5 Select "Create a new Open Directory domain," and click Next.

6 In the Directory Administrator pane, deselect the checkbox "Remember this password in my keychain."



7 Configure a password.

If your server is not accessible from the Internet, in the Directory Administrator pane, enter **diradminpw** in the Password and Verify fields.

Of course, in a production environment, you should use a secure password and consider using an account name different from the default "diradmin" so that it is more difficult for unauthorized people to guess the username and password combination.

8 Click Next.

9 In the Organization Information pane, enter the appropriate information.

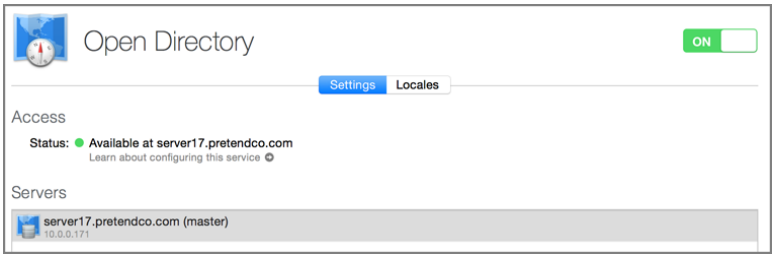
If the following fields do not already contain the information shown, enter it, and click Next.

- Organization Name: **Pretendco Project *n*** (where ***n*** is your student number)
- Admin Email Address: **ladmin@server*n*.pretendco.com** (where ***n*** is your student number)

10 View the Confirm Settings pane, and click Set Up.

The Server app displays its progress in the lower-left corner of the Confirm Settings pane.

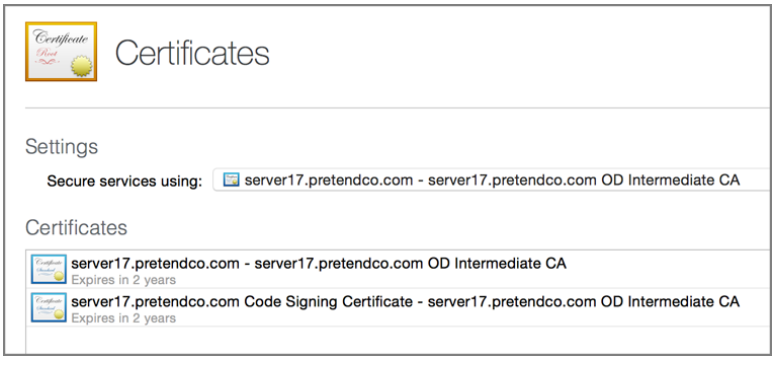
When it has completed the configuration, the Server app displays the Settings tab of the Open Directory pane, with your server listed as the master in the Servers list.



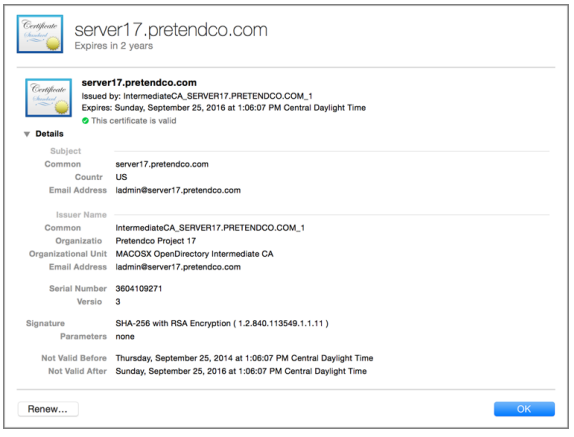
Inspect the OD Certificates

Inspect the certificates that the Server app automatically created.

1. In the Server app sidebar, select Certificates.
2. Click the Action (gear icon) pop-up menu, and choose Show All Certificates.
3. Confirm that the "Secure services using" pop-up menu is no longer set to a self-signed certificate but rather a certificate signed by your server's OD intermediate CA.



4. Confirm that the self-signed certificate is no longer listed in the Certificates field.
5. Double-click the certificate with your server's host name, signed by your OD intermediate CA (the first entry in the Certificates field).



6 Confirm that in the Issuer Name section, the first field (which is the common name) has a value made up of the following strings:

- "IntermediateCA_"
- Your server's host name in all capital letters
- "_1"

7 Click OK to close the certificate information pane.

8 Double-click the code signing certificate (the second entry in the Certificates field).

9 Confirm that this is also issued by your OD intermediate CA.

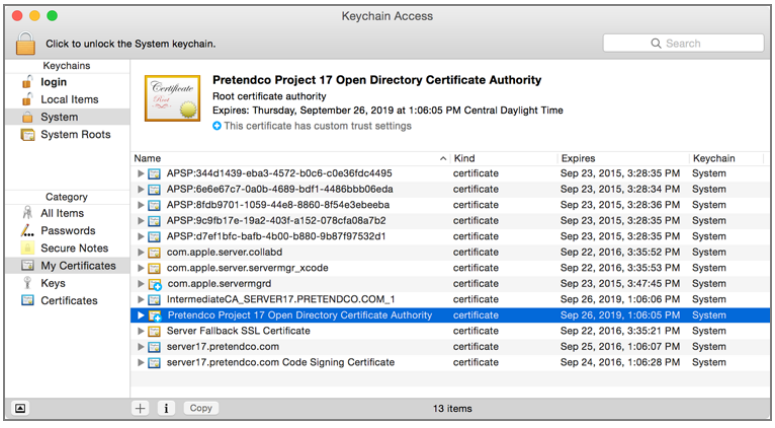
Use Keychain Access to inspect your OD CA, your OD intermediate CA, and the two signed certificates.

1 On your server, use a Spotlight search to open Keychain Access.

2 In the Keychains column, select System.

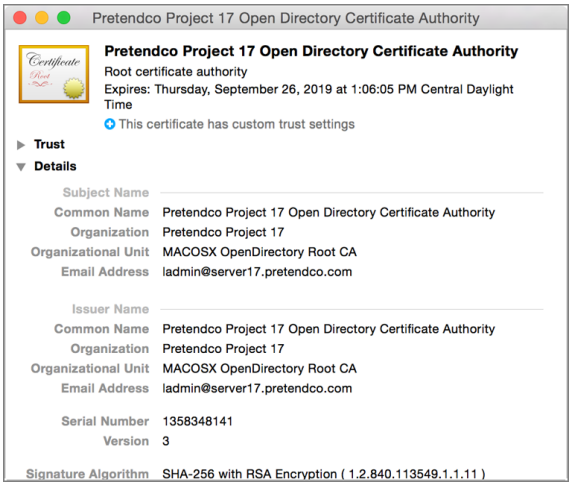
3 In the Category column, select My Certificates.

4 Select your OD CA. Its name is Pretendco Project *n* Open Directory Certificate Authority (where *n* is your student number).



5 Double-click your OD CA to examine it.

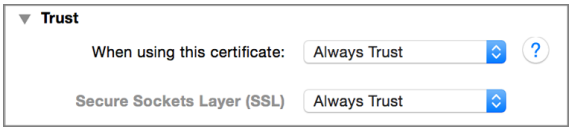
6 Confirm that the second line of text identifies it as "Root certificate authority" and that the Subject Name information matches the Issuer Name information.



7 Note that the certificate's color is bronze, which signifies that it is a root certificate.

8 Click the Trust disclosure triangle to display more details.

9 Confirm that your server is set to always trust this certificate.

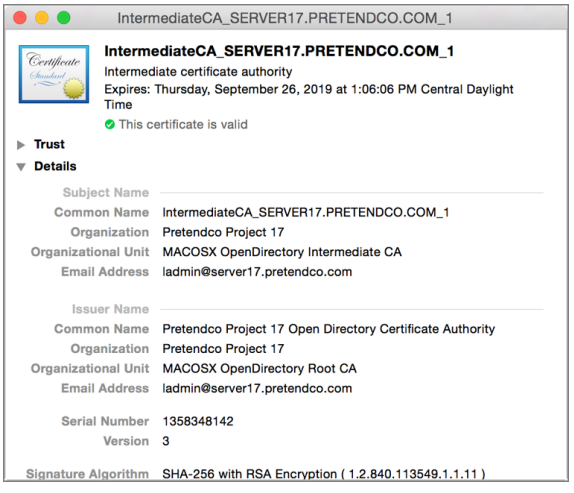


10 Close your OD CA.

11 Double-click your OD intermediate CA.

12 Confirm that its second line of text identifies it as "Intermediate certificate authority." Because your server trusts your OD CA and your OD CA signed this intermediate CA, this certificate is marked as valid with a green checkmark.

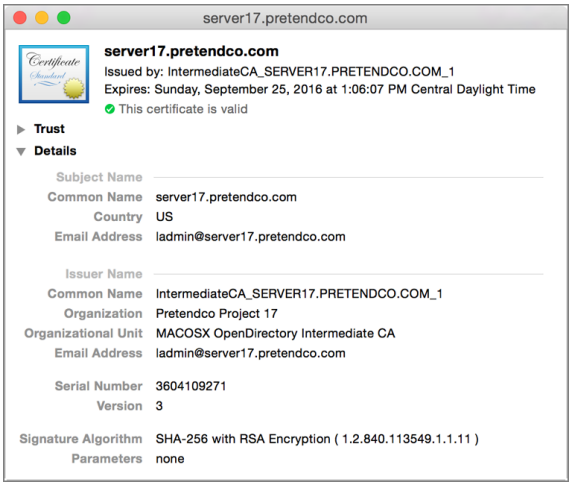
Note that the color of the certificate is blue, which signifies that it is an intermediate or leaf certificate.



13 Close your OD intermediate CA.

14 Double-click the certificate with your server's host name.

15 Confirm that the second line of text indicates that it is signed by your OD intermediate CA. Your server is configured to trust your OD CA, which signed your OD intermediate CA, which signed this certificate, so it is marked as valid with a green checkmark.



16 Double-click your code signing certificate, inspect it, and close it.

17 Quit Keychain Access.

In this exercise, you configured your server to be an Open Directory master. The Server app automatically configured a new OD CA, intermediate CA, and two new certificates; it removed your server's old default self-signed certificate, and it updated services to use the certificate signed by the intermediate CA. It automatically configured your server to trust its own OD CA, which means that your server also trusts the OD intermediate CA and the two other certificates that are signed by the OD intermediate CA.

Exercise 4.3
Configure Your Administrator Computer to Trust an SSL Certificate

Prerequisite

- "Exercise 4.2 Configure an Open Directory Certificate Authority" on page 152

NOTE ▶ If you obtained a certificate from a widely trusted CA, you do not need to perform this exercise.

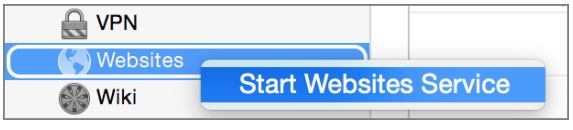
In a production environment, it is best to use a valid SSL certificate that's been signed by a trusted CA. If that isn't possible, you should configure your users' computers and devices to trust your server's certificate so that your users do not get into the habit of configuring their devices to trust unverified SSL certificates.

This lesson shows you how to configure an individual computer to trust your server's OD CA; it is beyond the scope of this exercise to show you how to replicate the end result on multiple computers and devices.

Turn On the Web Service Temporarily

Turn on your server's Websites service so you can quickly access the SSL certificate your server's services use.

1. In the Server app sidebar, Control-click Websites, and then choose Start Websites Service.



Visit Your Server's Website Protected by SSL

In this exercise, you will use your administrator computer and confirm that you are using your server's DNS service; otherwise, you will not be able to connect to its web service using its host name. Then you'll open Safari to your server's default HTTPS website. Finally, you'll configure your administrator computer to trust the SSL certificate.

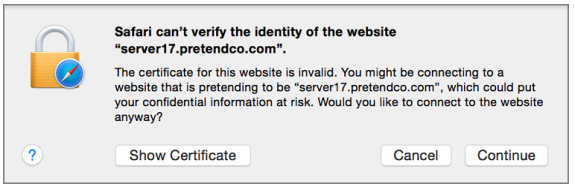
1. On your administrator computer, open System Preferences.
2. Open the Network pane.
3. Select the active network service, and confirm that your server's IP address is listed for the DNS Service value.

   If you are using Wi-Fi, you need to click Advanced and then click the DNS tab to view the DNS Service value.
4. Quit System Preferences.

5 On your administrator computer, open Safari, and in the Address and Search field, enter **https://server*n*.pretendco.com** (where ***n*** is your student number).

6 Press Return to open the page.

Your certificate is not signed by a CA that your administrator computer is configured to trust, so you'll see the message that Safari can't verify the identity of the website.

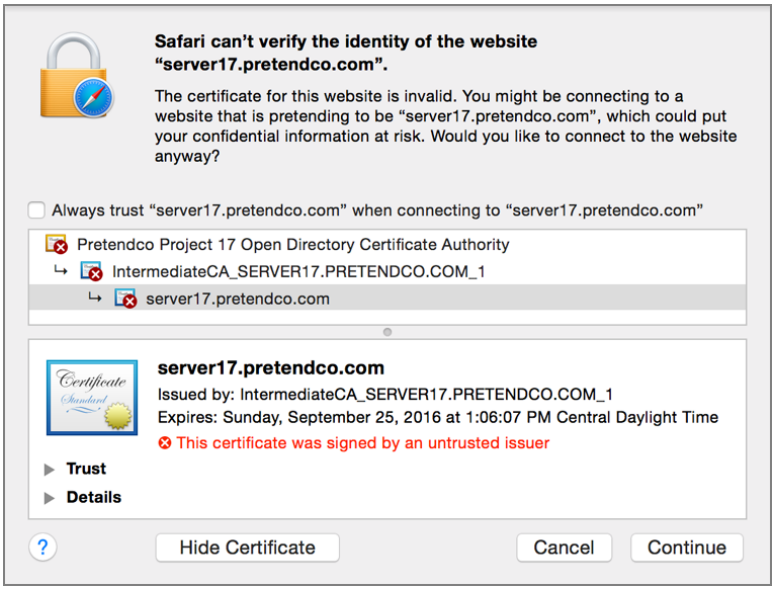


Configure Your Administrator Computer to Trust This SSL Certificate

Once you see the dialog that Safari can't verify the identity of the website, you can click Show Certificate and configure the currently logged-in user to trust the SSL certificate used by the website.

1 Click Show Certificate.

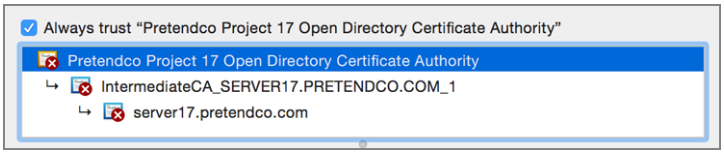
2 Note that the certificate with your server's host name is marked in red, "This certificate was signed by an untrusted issuer."



3 In the certificate chain, select your OD CA.

4 Click the Details disclosure triangle, and inspect the details.

5 Select the checkbox "Always trust 'Pretendo Project *n* Open Directory Certificate Authority'" (where *n* is your student number).



6 Click Continue.

7 Provide your login credentials, and click Update Settings.

This updates the settings for the currently logged-in user; this does not affect any other user on this computer.

8 Confirm the Safari Address and Search field displays a lock icon, which indicates that the page was opened using SSL.

9 Keep Safari open for the next section of this exercise.

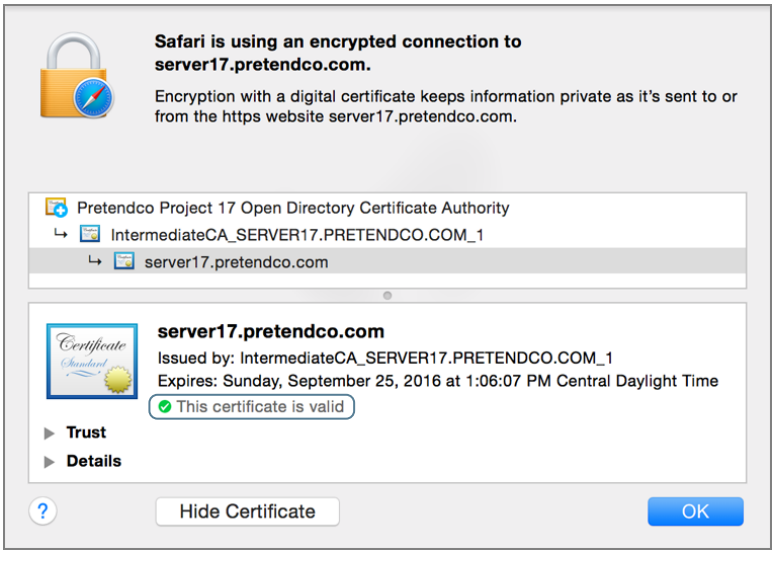
Confirm That Your Mac Trusts the SSL Certificate

To view the SSL certificate the Websites service is using, perform the following steps.

1 In the Safari Address and Search field, click the lock icon.

2 In the pane that informs you that Safari is using an encrypted connection, click Show Certificate.

3 Confirm that the certificate is listed as valid with a green checkmark.



4 Press Command-Q to quit Safari.

You confirmed that the Websites service uses the SSL certificate you configured in the previous exercise. You confirmed that by trusting a CA, you trust a certificate that was signed by an intermediate CA that was signed by the CA (at least for the currently logged-in user).

Exercise 4.4
Clean Up

To ensure that the rest of the exercises are consistent, turn off the Websites service.

1. In the Server app sidebar, select the Websites service, and click Off to turn the service off.
2. Confirm that no green status indicators appear next to the Websites service.

   This indicates that the service is off.

You are ready to complete the tasks of any other lesson's exercises.

This page intentionally left blank

Index

B

C

D

E

F

G

J

K

L

M

Q

R

S

T

U

V

W

X

Y

Z