



The Mac OS X Directory Services 10.6 Exam Skills Assessment Guide

To Register

This exam is offered at Apple Authorized Training Centers and Prometric Testing Centers worldwide. For information on exam registration, please visit training.apple.com/certification.

To learn more about Apple training offerings that can help you prepare for certification, please visit training.apple.com/courses.

The Mac OS X Directory Services 10.6 Exam (Prometric exam no. 9L0-624) is a computer-based test offered at Apple Authorized Training Centers and Prometric Testing Centers.

The exam is one of three required exams in the Apple Certified Specialist - Directory Services 10.6 certification track (ACS - DS). You must pass this exam, and also earn Apple Certified Technical Coordinator 10.6 certification by passing the Mac OS X Support Essentials 10.6 Exam and the Mac OS X Server Essentials 10.6 Exam to become ACS - DS 10.6 certified.

The exam is one of four required exams in the Apple Certified System Administrator (ACSA) 10.6 certification track. You must pass this exam, the Mac OS X Server Essentials 10.6 Exam, the Mac OS X Deployment 10.6 Exam, and the Mac OS X Security and Mobility 10.6 Exam to become ACSA 10.6 certified.

You may take up to two hours to complete the exam, which consists of 78 multiple-choice questions that are based on the objectives listed in this guide.

The score required to pass is 68 percent. Five demographic questions are presented but are not scored.

To prepare for the exam, read through the objectives in this guide to determine which areas you need to review. The primary reference source for this exam is the book: *Apple Training Series: Mac OS X Directory Services v10.6* (Peachpit 2010).

You will not have access to any resources or references during the exam. Please note that the exam is based on Mac OS X and Mac OS X Server version 10.6. All references to Mac OS X and Mac OS X Server, unless otherwise stated, refer to version 10.6.

The number of test questions drawn from each knowledge area is indicated below. Please note that although this guide divides the objectives into eight knowledge areas, questions are presented randomly during the exam. Also note that UNIX commands and processes are shown in `monospace font` in the exam. Finally, when the term “by default” is used in the exam, it means using only the settings or values assigned automatically by the operating system or application, without any custom configuration.

Accessing the Local Directory Service

This topic has 15 items, drawn from the following objectives:

- Without references, define the following terms: node, record, attribute.
- Without references, identify tools available to view DNS information, including `dig` and `dscacheutil`.
- Without references, list the tools to create and edit local user accounts, including Workgroup Manager, `dsc1`, and `vipw`.
- Without references, identify `dsc1` and Workgroup Manager as the tools used to edit a live Directory Service database.
- Without references, identify `dsc1` and a text editor as tools which can operate on a database without the `DirectoryService` daemon running.
- Without references, list the `master.password` and group flat files that hold information for BSD/local node.
- Without references, list the location of files that hold information for the `/Local/Default` node.
- Given a property list file containing a user record, identify the attributes in the record file and their values.
- Given a property list file containing a user record for a managed user, identify the attributes in the record file and their values.
- Without references, explain the function of the key user attributes in a BSD flat file: name, password, uid, gid, gecos, home_dir, shell.
- Given a standard installation of Mac OS X v10.6 and Workgroup Manager, create a user record in the `/Local/Default` node.
- Given `dsimport` in a standard installation of Mac OS X v10.6, create a user record in the `/Local/Default` node.
- Given a standard installation of Mac OS X v10.6, create a user record in the `/Local/Default` node by copying and modifying a record file.
- Given a text editor in a standard installation of Mac OS X v10.6, modify an existing attribute in a user record in `/Local/Default`.
- Given `dsc1` in a standard installation of Mac OS X v10.6, modify an existing attribute in a user record in `/Local/Default`.
- Given a text editor in a standard installation of Mac OS X v10.6, create a user record in the `/BSD/Local` node.
- Given a text editor in a standard installation of Mac OS X v10.6, modify an existing user in a user record in `/BSD/Local`.
- Without references, list `dsc1`, Workgroup Manager, and text editors as tools available to create and edit local group accounts.
- Given an XML file for a group record from the `/Local/Default` node, identify any group attributes in the records contained in the XML file.
- Given a BSD group file, identify any group attributes in the file.
- Without references, identify `dseditgroup` as a tool to create and edit groups in the `/Local/Default` directory.
- Without references, identify a text editor as a way to create and edit groups in `/BSD/Local`.
- Given a standard installation of Mac OS X v10.6 and Workgroup Manager, create groups in `/Local/Default`.

- Given a standard installation of Mac OS X v10.6 and Workgroup Manager, edit groups in `/Local/Default`.
- Given a standard installation of Mac OS X v10.6 and `dseditgroup`, create groups in `/Local/Default`.
- Given a standard installation of Mac OS X v10.6 and `dseditgroup`, modify the groups in the `/Local/Default` directory.
- Given a standard installation of Mac OS X v10.6 and `vipw`, create groups in the `/BSD/Local` directory.
- Given a standard installation of Mac OS X v10.6 and a text editor, edit groups in the `/BSD/Local` directory.
- Without references, identify the advantages of `dseditgroup` over `dsccl` for creating and editing groups.
- Without references, identify the location of Directory Service logs.
- Without references, describe how to enable detailed logging on the `DirectoryService` process.
- Without references, identify `kill` and `killall` as tools to signal a process.
- Without references, identify `dsccl` as a tool to test a directory services search path.
- Given a standard installation of Mac OS X v10.6, send a signal to a process.
- Given a standard installation of Mac OS X v10.6, enable detailed logging of the `DirectoryService` process.
- Given Terminal on a Mac OS X v10.6 computer, display the meaning of a numerical Directory Service error.
- Given a standard installation of Mac OS X v10.6, identify issues caused by user name collision with two directories.
- Given a standard installation of Mac OS X v10.6, resolve issues caused by user name collision with two directories.
- Given Terminal on a Mac OS X v10.6 computer, test user authentication.
- Without references, state the command-line utility to display the error message for a given `DirectoryService` error number.

Accessing an Open Directory Server

This topic has 11 items, drawn from the following objectives:

- Without references, list potential security issues when binding to a directory server.
- Without references, define the following term: trusted binding.
- Without references, list the requirements for trusted binding.
- Given Directory Utility, a Mac OS X v10.6 computer, and an Open Directory server, configure the Mac OS X computer to bind to the directory server with trusted binding.
- Given Directory Utility and a Mac OS X v10.6 computer bound to an Open Directory server, configure the computer to encrypt all traffic with the directory server.
- Without references, describe the stages in the binding process.
- Without references, describe how Mac OS X auto-configures the search base and schema of Open Directory servers.

- Without references, identify the client log files that are written to during binding.
- Without references, define the following terms: search base, authentication search policy, and contacts search policy.
- Without references, describe the difference between the search policies: Automatic, Local, Custom.
- Without references, explain how multiple directory nodes are searched when `DirectoryService` searches for a user.
- Without references, identify Directory Utility as a tool to configure Mac OS X v10.6 to use the local BSD flat files.
- Without references, identify Directory Utility as a method to configure a custom path search policy for directory services.
- Without reference, identify the client log files that are written to during a login attempt.
- Without reference, describe the login process for a network user.
- Without references, identify `klist` as a tool for verifying Kerberos authentication.
- Given `dsc1` in a standard installation of Mac OS X v10.6, verify that Mac OS X is able to access user accounts via the `/Search` path.
- Given the Kerberos application and a Mac OS X v10.6 computer, verify Kerberos authentication.
- Given the Finder on a Mac OS X v10.6 computer, verify home directory access.
- Given the Console application and a Mac OS X v10.6 computer, display the directory service log files.
- Given a list of Kerberos tickets as displayed by the Kerberos application on a Mac OS X v10.6 computer, identify which Kerberos tickets came from a client's local KDC.

Accessing a Third-Party LDAP Service

This topic has 11 items, drawn from the following objectives:

- Given an OpenLDAP server, create all records necessary for mounting a network home directory when a user logs into a Mac OS X computer bound to the server.
- Without references, define the following terms: schema, object class.
- Without references, list authentication methods available for `loginwindow`.
- Without references, describe the schema for a mount record and the schema for managed preferences (MCX) records.
- Without references identify specific mapping issues with an eDirectory server.
- Without references, identify Directory Utility as a tool to configure client computers to augment LDAP login information with managed client information from Open Directory server.
- Without references, identify Directory Utility as a tool to configure client computers to augment LDAP login information with mount information from an Open Directory server.
- Without references, identify Directory Utility as a tool to configure client computers to augment LDAP login information with local static map information.
- Without references, identify Kerberos command-line tools included in a standard installation of Mac OS X v10.6.

- Given a standard installation of Mac OS X v10.6, identify the location of the record and attribute mappings.
- Given a list of LDAP records and attributes, and Directory Utility, configure the LDAP binding to map the Mac OS X records and attributes to available LDAP records and attributes.
- Given Directory Utility and a standard installation of Mac OS X v10.6, configure a Mac OS X client computer to supplement LDAP login information with mount information from an Open Directory server.
- Given Directory Utility and a standard installation of Mac OS X v10.6, configure a Mac OS X client computer to augment LDAP login information with local static map information.
- Without references, describe how to configure a Mac OS X client computer to authenticate to a third-party KDC using Kerberos command-line tools in a standard installation of Mac OS X v10.6.
- Without references, describe the effects of using static mappings for an LDAP configuration in Directory Utility.
- Given an LDAP configuration in Directory Utility on a Mac OS X v10.6 computer, modify the mappings to include static or dynamic values.
- Without references, describe the process of user authentication at the login window.
- Without references, describe the process the system uses to apply managed preference (MCX) policies at login.
- Without references, describe the process the system uses to locate the user's home folder at login.

Accessing an Active Directory Service

This topic has 3 items, drawn from the following objectives:

- Without references, define the following terms: computer trust account, domain, forest.
- Without references, describe Active Directory packet signing and encryption.
- Without references, identify Account Preferences, Directory Utility, and `dsconfigad` as tools to bind to an Active Directory server.
- Given the Directory Utility tool in a standard installation of Mac OS X v10.6, bind to an Active Directory server.
- Given the `dsconfigad` command-line tool in a standard installation of Mac OS X v10.6, bind to an Active Directory server.
- Without references, describe the stages in the binding process to an Active Directory server.
- Without references, list client log files that are written to during binding.
- Given Terminal in a standard installation of Mac OS X v10.6, perform DNS lookups to verify the records required by Active Directory.
- Given `rm` in a standard installation of Mac OS X v10.6 which is bound to an Active Directory server, remove all files containing AD binding information.
- Given a log file, identify any log entries related to an unsuccessful binding.

Configuring Open Directory Server

This topic has 14 items, drawn from the following objectives:

- Given Server Admin, describe the LDAP options available when configuring Open Directory.
- Without references, compare and contrast the “Create...”, “Import...”, and “Manual...” configuration options that are available when you initially configure a Mac OS X Server using Server Assistant.
- Without references, list the methods to upgrade a Mac OS X Server.
- Without references, describe the advantages and drawbacks of the various methods that are available to upgrade a Mac OS X Server Open Directory master.
- Without references, identify `changeDirData.pl` as a tool to perform batch updates on an OpenLDAP database.
- Without references, identify `slap_config` as a tool to promote an Open Directory master.
- Without references, identify Server Admin as a tool to tune Open Directory security and performance options.
- Given the `slap_config` tool in a standard installation of Mac OS X Server v10.6, promote an Open Directory master.
- Given the Server Admin application in a standard installation of Mac OS X Server v10.6, tune Open Directory server for security and performance.
- Given a Mac OS X v10.5 Open Directory master, upgrade it to Mac OS X v10.6 without losing any directory information.
- Without references compare and contrast the two types of tools that come with OpenLDAP, including those that operate directly on the LDAP databases and those that go through the LDAP protocol.
- Without references, define the following term: directory access controls (DACs).
- Without references, describe how Mac OS X Server enforces tiered administration.
- Without references, define the following acronym: LDIF.
- Without references, compare and contrast the possible uses of an LDIF file with those of an XML file exported from Workgroup Manager.
- Without references, identify command-line OpenLDAP tools to add users to an Open Directory server.
- Without references, compare the command-line tools, `dsimport`, `ldapadd`, and `slapadd`.
- Without references, identify the log files used by Open Directory when promoting Mac OS X Server to an Open Directory master.
- Without references, describe the tools, processes, and files involved in the promotion of an Open Directory server.

Configuring Open Directory Replicas

This topic has 10 items, drawn from the following objectives:

- Without references, describe the topology of Open Directory replication.
- Without references, identify the maximum number of records supported in an Open Directory master.

- Without references, identify the amount of load an Open Directory client places on an Open Directory master.
- Given an existing Open Directory master and the Server Admin application in a standard installation of Mac OS X Server v10.6, create an Open Directory replica of the Open Directory master.
- Given an Open Directory replica and a standard installation of Mac OS X v10.6, bind the Mac OS X client computer to the Open Directory replica so that it will fail over to another Open Directory server in the event of a failure of the Open Directory replica.
- Given a failed Open Directory master and an Open Directory replica, recover from a failure of the Open Directory master.
- Given a network topology, plan an Open Directory system.
- Describe the tools, processes, and files involved in creating an Open Directory replica.
- Given the Console application in a standard installation of Mac OS X v10.6, review the logs involved in the Open Directory replication process.

Connecting Mac OS X Server to Open Directory

This topic has 10 items, drawn from the following objectives:

- Given the Server Admin application in a standard installation of Mac OS X v10.6, bind the server to an Open Directory master.
- Given the Server Admin application in a standard installation of Mac OS X v10.6, bind the server to an Open Directory Kerberos realm.
- Without references, list services which can use network user accounts for access control.
- Without references, describe the use of the UUID (also known as a GUID) to identify users in an Open Directory system.
- Given the Workgroup Manager application and a Mac OS X server bound to an Open Directory master, view users on the parent Open Directory master.
- Given Server Admin a Mac OS X server bound to an Open Directory master, assign file system privileges of local files to users and groups on the master.
- Without references, describe the tools, processes, and files used in joining a Kerberos realm.
- Without references, describe how keytabs and principals are created when joining a Kerberos realm with Server Admin.
- Without references, define the following terms: keytab, principal.
- Given the Console application on a Mac OS X server which has joined an Open Directory master, view the log files that document the binding process.
- Given Directory Utility on a Mac OS X server which has joined an Open Directory master, revert the server to a pre-join state and join again.
- Without references, explain how a TGT is generated and used for authentication.
- Without references, explain how a Kerberos service ticket is generated and used for authentication.
- Without references, define the following terms: service principal, user principal.

- Given a Mac OS X v10.6 computer bound to an Open Directory master, verify the service principals installed on the server are valid.
- Without references, describe how Kerberos generates a ticket-granting ticket.
- Without references, describe how service tickets are used by user authentication in Kerberos.
- Without references, describe how setting up and using a network time server can help avoid some Kerberos authentication issues.

Integrating Mac OS X Server with Other Systems

This topic has 4 items, drawn from the following objectives:

- Without references, define the following term: augmented record.
- Without references, describe how augmented records can extend the schema of a third-party directory server.
- Given the Open Directory Administration Guide, list the object classes Open Directory adds to the standard LDAP schema.
- Given a Mac OS X computer bound to multiple directories, identify which directory will be used for authentication.
- Given a Mac OS X computer bound to multiple directories, identify which directory will be used for identification.
- Given a computer with Mac OS X Server installed and an existing directory server, configure the Mac OS X server to augment the user records of the existing directory server.
- Without references, describe the role of keytabs in Kerberos authentication.
- Given Kerberos tools in a standard installation of Mac OS X Server v10.6 configured as an Open Directory master, create service keys for a third-party Kerberos service.

For More Information

Please visit training.apple.com for more information about all Apple training courses and certification programs.