VMS Software

VSI OpenVMS ACME LDAP Installation and Configuration Guide

Operating System and Version: VSI OpenVMS IA-64 Version 8.4-1H1 or higher
VSI OpenVMS x86-64 Version 9.2-3 or higher

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

VSI OpenVMS ACME LDAP Installation and Configuration Guide

Legal Notice

Table of Contents

Preface

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

2. Intended Audience

This document is intended for OpenVMS system administrators. For more information about system security, see the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

3. Related Documents

The following resources can be referred for more information:

- SYS$HELP:ACME_DEV_README.TXT
- The sections "Enabling External Authentication" [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#EXTERNAL_AUTH] and "Authentication and Credentials Management Extensions (ACME) Subsystem" [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#auth_credentials] in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].
- *VSI OpenVMS System Services Reference Manual: A-GETUAI* [https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/]

4. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at https://docs.vmssoftware.com.

5. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: `<docinfo@vmssoftware.com>`. Users who have VSI OpenVMS support contracts through VSI can contact `<support@vmssoftware.com>` for help with this product.

6. Typographical Conventions

The following conventions are used in this manual:

Convention	**Meaning**
...	A horizontal ellipsis in examples indicates one of the following possibilities: ● Additional optional arguments in a statement have been omitted.

Convention	Meaning
	● The preceding item or items can be repeated one or more times. ● Additional parameters, values, or other information can be entered.
⋮	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one. In installation or upgrade examples, parentheses indicate the possible answers to a prompt, such as: `Is this correct? (Y/N) [Y]`.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
italic type	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (**`/PRODUCER=`*`name`***), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates the name of a routine, the name of a file, or the abbreviation for a system privilege.
`monospace type`	Monospace type indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language.
`bold monospace type`	Bold monospace type indicates a DCL command or command qualifier.
–	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.

Chapter 1. Overview

Lightweight Directory Access Protocol (LDAP) is combined with the Authentication and Credentials Management Extension (ACME) authentication mechanism to provide a solution to customers to manage all accounts in a centralized directory.

The ACME LDAP agent provided with OpenVMS provides "simple bind" authentication during login using an LDAP-compliant directory server. In this authentication method, users enter their LDAP entry name and password. An LDAP attribute is configured, which is used to match the entered username so that the authentication can take place. The following sections provide information on how to install and configure the standard ACME LDAP agent.

Secure Socket Layer (SSL)/Transport Layer Security(TLS) LDAP communication is supported to prevent cleartext passwords from being exposed over the network. Dedicated SSL port and the startTLS operation over the standard port are supported.

Chapter 2. Enabling and Configuring the ACME LDAP Agent

2.1. Prerequisites

- You must be running either OpenVMS IA-64 Version 8.4-1H1 or later, or OpenVMS x86-64 Version 9.2-3 or later.
- You must install the SYS$ACM enabled (ACMELOGIN) LOGINOUT.EXE and SETP0.EXE images.

  For more information, see the SYS$HELP:ACME_DEV_README.TXT file.

2.2. General Setup

You must first configure and populate your LDAP directory server with user entries. The ACME LDAP agent is configured by performing the following steps:

1. Enable the SYS$ACM (ACMELOGIN) enabled LOGIN.
2. Set up the LDAP Persona Extension.
3. Configure the ACME LDAP Agent.
4. Start the ACME LDAP Agent.

2.3. Enabling the SYS$ACM (ACMELOGIN) Enabled LOGIN

To enable the SYS$ACM enabled LOGIN (previously known as ACMELOGIN) and ACME LDAP, simply run the command file SYS$MANAGER:SYS$SWITCH_LOGIN.COM.

Note

The procedure is different for OpenVMS IA-64 V8.4-1H1. For more detailed steps on installation, see SYS$HELP:ACME_DEV_README.TXT.

When the ACME LDAP agent is enabled, proceed to the next section, Section 2.4, “Setting Up the LDAP Persona Extension”.

2.4. Setting Up the LDAP Persona Extension

To set up the persona extension on OpenVMS IA-64 Version 8.4-1H1 or later, complete the following steps:

1. Add an entry for the persona extension image to the systems images list as follows:

```
$ MCR SYSMAN
SYSMAN> SYS_LOADABLE ADD LDAPACME LDAPACME$EXT
SYSMAN> exit
```

2. Generate a new system images data file using the following command:

   ```
   $ @SYS$UPDATE:VMS$SYSTEM_IMAGES.COM
   ```

3. Reboot the system using the following command:

   ```
   $ @SYS$SYSTEM:SHUTDOWN
   ```

   During reboot, an error message will appear if the persona extension image is not loaded. If the error message is not displayed, it means that the image is loaded as required.

To set up the persona extension on OpenVMS x86-64 Version 9.2-3 or later, complete the following steps:

1. Add an entry for the persona extension image to the systems images list as follows:

   ```
   $ MCR SYSMAN
   SYSMAN> SYS_LOADABLE ADD LDAPACME LDAPACME$EXT
   SYSMAN> exit
   ```

2. Generate a new system images data file using the following command:

   ```
   $ @SYS$UPDATE:VMS$SYSTEM_IMAGES.COM
   ```

3. Reconfigure the node memory disk using the following command procedure:

   ```
   $ @SYS$UPDATE:SYS$MD.COM
   ```

   This procedure creates a new system bootable memory disk that contains the LOGINOUT and LDAP components necessary for external authentication processing.

4. Reboot the system to load the new memory disk using the following command:

   ```
   $ @SYS$SYSTEM:SHUTDOWN
   ```

After setting up the LDAP persona extension, you can proceed towards configuring your ACME LDAP agent. See Section 2.5, “Configuring the ACME LDAP Agent”.

2.5. Configuring the ACME LDAP Agent

Configuration of the ACME LDAP agent involves the following steps:

1. Edit the LDAP configuration file
2. Start the ACME LDAP agent

The attribute used for usernames is specified by the `login_attribute` directive in your ACME LDAP INI configuration file. For more information about `login_attribute`, see Table 2.1, “LDAP Configuration Attributes”.

The ACME LDAP agent searches this attribute on the directory server for matching usernames (entered at the "Username" prompt during login). The search is done in the set of LDAP entries below the point in your directory tree specified by the `base_dn` directive.

The username (entered at the "Username" prompt during login) is mapped to the username in the SYSUAF.DAT file. Global and local mappings are also supported. For more information on global and local mapping, see Chapter 3, *Global and Local Mapping*.

OpenVMS-specific information, such as privileges, identifiers, and so on, is taken from the SYSUAF.DAT file.

A user scenario on configuring the ACME LDAP agent and a sample login are provided in Chapter 4, *User Scenario of Configuring an OpenVMS ACME LDAP Agent*.

2.5.1. Editing the LDAP Configuration File

To edit the ACME LDAP INI file, perform the following steps:

1. Make a copy of SYS$STARTUP:LDAPACME$CONFIG-STD.INI_TEMPLATE and rename it to any file name of your choice. For example, the following command renames the file to SYS$STARTUP:LDAPACME$CONFIG-STD.INI:

   ```
   $ COPY SYS$STARTUP:LDAPACME$CONFIG-STD.INI_TEMPLATE
   SYS$STARTUP:LDAPACME$CONFIG-STD.INI
   ```

2. Edit SYS$STARTUP:LDAPACME$CONFIG-STD.INI to specify the directives that correspond to your requirements.

   For descriptions of the directives present in the LDAPACME$CONFIG-STD.INI file, refer to the following table.

Table 2.1. LDAP Configuration Attributes

Directive	Description

server	This is a mandatory directive.
	Use the `server` directive to provide the IP address (or DNS host name) for your directory server.
	You can specify one or more redundant servers by providing spaces between the server name or IP address. For example:
	`server = test1.testdomain.com test2.testdomain.com`
	`server = test1.testdomain.com test2.testdomain.com test3.testdomain.com`
	Initially, the ACME LDAP agent tries to connect to the first server. If the connection to the first server fails, the second server is tried for connection. If the second server connection also fails, the next set of servers is tried in sequence, until the last server in the list.
	Note the following while using redundant servers:
	● The `base_dn`, `bind_dn`, and `bind_password` directive values must be the same on all the redundant directory servers. The user records getting authenticated using the ACME LDAP must also be present on all the directory servers.

Directive	Description
	● Set the `bind_timeout` directive when using redundant multiple servers. This ensures that the ACME LDAP tries to connect to all the redundant servers before the user session times out. ● If you have provided the Certificate Authority's (CA) public key (`ca_file` directive) and the public keys are different, provide all the public keys in the same `ca_file`. For more information, see the `ca_file` directive.
port	This is a mandatory directive. The port that your directory server is listening for. Defaults to the standard port 389 (or 636 for SSL/TLS).
login_attribute	This is a mandatory directive. The LDAP schema attribute that contains the username for login purposes. This is often specified as 'uid', but may be different in your configuration.
password_type	Select one of the following: ● standard (default) ● active-directory If this directive is not specified, the **`SET PASSWORD`** command fails.
password_update	Applies only when `password_type=standard` is set. Some directory servers require the old password to be supplied when changing userPassword attribute; others do not. Select one of the following: ● replace (default) ● remove-and-add
base_dn	The LDAP users are stored in a tree structure in your directory server. The `base_dn` directive is the distinguished name of a tree element on the directory server. All the user entries must be present under this tree element as sub-tree elements. The ACME LDAP will search for matching entries within this sub-tree based on the attribute specified by `login_attribute`. (See the `scope` directive.)
scope	Controls the depth of the search beneath the base_dn. Valid keywords are: ● sub – searches the base entry and all entries at all levels below the base entry ● one – searches all entries at one level below the base entry ● base – searches only the base entry If you are not sure about the keyword to be used, you can use "sub" as the keyword.

Directive	Description
filter	This directive is optional. Search filter for limiting the objects that will be searched for users in the LDAP tree. Defaults to `objectclass=*`.
bind_dn	The distinguished name (DN) of a user account (directory entry) that is granted "search" permission through the directory sub-tree specified by `base_dn`. The `bind_dn` directive, along with the `bind_password` directive, is used to bind to your directory servers, before searching for users on the directory servers. Some directory servers will not allow the ACME LDAP agent to bind to them by default without `bind_dn` and `bind_password`. The `bind_dn` and `bind_password` directives must be specified in such cases. Some directory servers will support anonymous binds to happen and you do not have to provide the `bind_dn` and `bind_password` directives for working with these directory servers.
bind_password	The password for the directory DN specified by bind_dn.
bind_timeout	Use the `bind_timeout` directive if you are providing multiple redundant servers in the server directive. Each bind request to a directory server, by default, takes around 75 seconds (TCPIP default connection establishment timeout), if the directory server is not reachable. If there are multiple redundant servers, the user login session (for example, a TELNET session) expires (within approximately 30 seconds) before the ACME LDAP agent checks the list of all servers mentioned in the server directive. The `bind_timeout` directive takes a timeout value in seconds for connecting to one directory server in the list of all servers mentioned in the server directive. For example, if you have two servers mentioned in the server directive and the `bind_timeout` directive is set to three seconds, the overall timeout period is around six seconds.
port_security	This is a mandatory directive. Specifies the method used to encrypt communications over the LDAP port. Possible values are "starttls" (the default), "ssl" (dedicated SSL port), or "none" (not recommended).
ca_file	This directive is optional. Specifies the file path of a PEM-format file containing the public key of the certificate authority that signed your directory server's public key.

Directive	Description

	The ACME LDAP agent checks this certificate file and whether it is connecting to the right directory server, when the `port_security` is set to "ssl" or "starttls".
	If this attribute is not used, the LDAP server's certificate is NOT verified.
	If there are redundant servers having different public key certificates, add the certificate information of all the servers into the same file:
	For example:
	`$ TYPE CACERT.PEM`
	`-----BEGIN CERTIFICATE-----`
	`.......`
	`server 1 public key certificate in base64 encoded format`
	`.......`
	`-----END CERTIFICATE-----`
	`-----BEGIN CERTIFICATE-----`
	`.......`
	`server 2 public key certificate in base64 encoded format`
	`.......`
	`-----END CERTIFICATE-----`
	`$`

mapping	Specifies whether the mapping is global or local. You are provided two options for this directive: ● Server ● Local For example: `mapping=server` indicates that global mapping is enabled for the user. `mapping=local` indicates the local mapping is enabled for the user. If the `mapping` directive is not used, mapping will be one-to-one.
mapping_attribute	This directive is applicable only for global mapping. Set this to the attribute on directory server that is used for user mapping. For example, the `mapping_attribute` directive can be referenced to the description attribute for the user in the directory server: `mapping_attribute=description` You can also use any newly created attribute on the directory server for mapping. The attribute should be an IA5 multi-valued string.

mapping_target	This directive is applicable only for global mapping. The `mapping_target` directive is searched in the value of the directory server's `mapping_attribute` field. For example:
	Let the LDAP INI file have:
	`mapping_attribute=description`
	`mapping_target= VMSUsers.vmssoftware.com`
	Let the description (field in directory server) be populated with:
	`VMSUsers.vmssoftware.com/jdoe`

Directive	Description
	The ACME LDAP agent then searches in `VMSUsers.vmssoftware.com/jdoe`, for a prefix of `VMSUsers.vmssoftware.com/` (with a forward slash (/) along with the `mapping_target`). The rest of the value, `jdoe`, is considered as the user name present in the SYSUAF.DAT file. If a multi-valued string attribute is used, the `VMSUsers.vmssoftware.com/jdoe` must be one of the array elements of the multi-valued string.
mapping_file	This directive is applicable only for local mapping. Set this to the complete path of the text database file to be searched for mapping users. A template file is available in SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT_TEMPLATE. This file includes the LDAP username and VMS username separated by a comma, where LDAP username is the name of the user in the domain (entered at the "Username" prompt during login). For information on how to populate and load the contents of the database file, see SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT_TEMPLATE.
domain	This directive is applicable for multi-domain support. Set this to the appropriate domain name.

3. Edit SYS$MANAGER:ACME$START.COM and define the following logical names:

   a. The LDAPACME$INIT logical name must contain the path name to the initialization for the ACME LDAP agent server. It can be defined with the following command:

   ```
   $ DEFINE/SYSTEM/EXECUTIVE LDAPACME$INIT -
   _$SYS$STARTUP:LDAPACME$CONFIG-STD.INI
   ```

   b. For cases of multi-domain support, create one configuration file for each domain. For example:

   - For the AMERICAS domain, create a configuration file as follows:

     ```
     SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_AMERICAS.INI
     ```

   - For the EMEA domain, create a configuration file as follows:

     ```
     SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_EMEA.INI
     ```

   Edit SYS$MANAGER:ACME$START.COM and define the LDAPACME$INIT logical name to point to all domain specific configuration files:

   ```
   $ DEFINE/SYSTEM/EXECUTIVE LDAPACME$INIT -
   _$SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_AMERICAS.INI, -
   _$SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_EMEA.INI
   ```

4. Remove the comment from the following line from SYS$MANAGER:ACME$START.COM:

```
$! @SYS$STARTUP:LDAPACME$STARTUP-STD                    ! LDAP
```

Important

The LDAPACME$INIT logical must be defined prior to starting the ACME LDAP agent. VSI recommends that you place this logical name in the command file SYS$MANAGER:ACME$START.COM before the SYS$STARTUP:LDAPACME$STARTUP-STD procedure executes.

5. Ensure that the LDAP configuration file and the LDAP local database mapping file are accessible for privileged users only. You can set the security of these files appropriately based on your security requirements. For example, the following command sets the accessibility of LDAPACME$CONFIG-STD.INI and LDAP_LOCALUSER_DATABASE.TXT files only for system users:

   ```
   $ SET SECURITY / PROTECTION = (system:"RWED", OWNER:"", GROUP:"", -
   WORLD:"") SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD.INI
   $ SET SECURITY / PROTECTION = (system:"RWED", OWNER:"", GROUP:"", -
   WORLD:"") SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE.TXT
   ```

2.5.2. Starting the ACME LDAP Agent

Restart the ACME_SERVER process using the following commands:

```
$ SET SERVER ACME/EXIT/WAIT
$ SET SERVER ACME/START=AUTO
```

Tip

You can place this command in your SYS$MANAGER:SYSTARTUP_VMS.COM procedure to have the ACME LDAP agent started automatically at boot.

2.6. Specifying EXTAUTH and VMSAUTH Flags on OpenVMS

For any user to be externally authenticated (via LDAP), the ExtAuth flag has to be set for the user account in the SYSUAF.DAT. When the ExtAuth flag is specified for a user account, the user is validated only externally using external authenticator (LDAP). If you want this user to be authenticated locally as well against the SYSUAF.DAT file, set VMSAuth flag for the user account in the SYSUAF.DAT file and use **/LOCAL** qualifier during login as described in the following section.

To set ExtAuth flag to the user, enter the following:

```
$ SET DEFAULT SYS$SYSTEM
$ MCR AUTHORIZE MODIFY <username> /FLAGS=(EXTAUTH,VMSAUTH)
```

A sample user profile is shown as follows:

```
$ SET DEF SYS$SYSTEM
$ MC AUTHORIZE
UAF> modify jdoe/flags=(EXTAUTH,VMSAUTH)
%UAF-I-MDFYMSG, user record(s) updated
UAF> sh jdoe

Username:  JDOE                                Owner:
```

```
Account:   TEST                             UIC: [201,2011] ([JDOE])
CLI:       DCL                              Tables: DCLTABLES
Default:   SYS$SYSDEVICE:[JDOE]
LGICMD:
Flags:  ExtAuth  VMSAuth
Primary days:    Mon Tue Wed Thu Fri
Secondary days:                      Sat Sun
No access restrictions
Expiration:             (none)    Pwdminimum:  6   Login Fails:   1
Pwdlifetime:         90  00:00    Pwdchange:      (pre-expired)
Last Login:             (none) (interactive),           (none) (non-interactive)
Maxjobs:           0 Fillm:        128 Bytlm:       128000
Maxacctjobs:       0 Shrfillm:       0 Pbytlm:           0
Maxdetach:         0 BIOlm:        150 JTquota:       4096
Prclm:             8 DIOlm:        150 WSdef:         4096
Prio:              4 ASTlm:        300 WSquo:         8192
Queprio:           4 TQElm:        100 WSextent:     16384
CPU:          (none) Enqlm:       4000 Pgflquo:     256000
Authorized Privileges:
  NETMBX       TMPMBX
Default Privileges:
  NETMBX       TMPMBX
UAF>
```

If your directory server is configured and your SYSUAF account is mapped with the user name on the directory server, you can now login to the system using ACME LDAP as the authentication agent as shown in the following example.

The password for user "jdoe" is validated against the password from directory server. Note that if the password in directory server is different from the password in the SYSUAF.DAT file, then the password on the SYSUAF.DAT file will be synchronized to the password on directory server. You can disable the password synchronization for a specific user or for all the users on the system. For more information on disabling the password synchronization, see the sections "Enabling External Authentication" [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#EXTERNAL_AUTH] and "Authentication and Credentials Management Extensions (ACME) Subsystem" [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/#auth_credentials] in the *VSI OpenVMS Guide to System Security* [https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security/].

```
$ TELNET 127.0.0.1
%TELNET-I-TRYING, Trying ... 127.0.0.1 %TELNET-I-
SESSION, Session 01, host 127.0.0.1, port 23 -TELNET-
I-ESCAPE, Escape character is ^]

Welcome to VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1

Username: jdoe
Password:
        VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1
        **** Logon authenticated by LDAP ****
    OpenVMS password has been synchronized with external password
```

In the following example, the user `jdoe` is validated against the SYSUAF.DAT file. Note that the user will not be mapped when the **`/LOCAL`** qualifier is provided during login. The username `jdoe` must be present in the SYSUAF.DAT file:

```
$ TELNET 127.0.0.1
%TELNET-I-TRYING, Trying ... 127.0.0.1 %TELNET-I-
SESSION, Session 01, host 127.0.0.1, port 23 -TELNET-
I-ESCAPE, Escape character is ^]

Welcome to VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1
```

```
Username: jdoe/local
Password:
   VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1
   Last interactive login on Tuesday, 1-DEC-2015 01:34:50.26
```

2.7. Examples of Configuration Files

Example 1 – Red Hat or Fedora Directory Server Configuration File

A sample configuration file using the Red Hat or Fedora directory server:

```
server = roux.vsi.com
port = 636
port_security = ssl
bind_dn = uid=acme-admin,ou=people,dc=acme,dc=mycompany,dc=com
bind_password = swordfish
base_dn = ou=people,dc=acme,dc=mycompany,dc=com
login_attribute = uid
scope = sub
ca_file = sys$manager:acme_ca.crt
```

Example 2 – Active Directory Configuration File

```
server = acme.mycompany.com
port = 636
port_security = ssl
password_type = active-directory
bind_dn = cn=acme-admin,cn=users,dc=acme,dc=mycompany,dc=com
bind_password = swordfish
base_dn = cn=users,dc=acme,dc=mycompany,dc=com
login_attribute = samaccountname
scope = sub
ca_file = sys$manager:acme_ca.crt

server = cssn-ddrs.testdomain.vmssoftware.com
port = 389
bind_dn = CN=query_account,CN=Users,DC=testdomain,DC=vmssoftware,DC=com
bind_password = welcome@123
base_dn = DC=testdomain,DC=vmssoftware,DC=com
scope = sub
port_security = none
password_type = active-directory

server = cssn-ddrs.Americas.vmssoftware.com
port = 389
bind_dn = CN=query_account,CN=Users,DC=Americas,DC=vmssoftware,DC=com
bind_password = welcome@123
base_dn = DC=Americas,DC=vmssoftware,DC=com
scope = sub
port_security = starttls
password_type = active-directory
domain = Americas

server = cssn-ddrs.testdomain.vmssoftware.com
```

```
port = 389
bind_dn = CN=query_account,CN=Users,DC=testdomain,DC=vmssoftware,DC=com
bind_password = welcome@123
base_dn = DC=testdomain,DC=vmssoftware,DC=com
scope = sub
port_security = starttls
password_type = active-directory
ca_file = sys$manager:cssn-ddrs.cer
```

You can configure the ACME LDAP agent to search multiple redundant directory servers for user authentication. This is helpful in a scenario where the first directory server is not reachable or active. As a result, the ACME LDAP agent tries to connect to a set of directory servers to authenticate the user.

In order to provide multiple redundant servers, the mandatory directives (such as `server` and `bind_timeout`) and the optional directive `ca_file` must be updated. For more information on the directives, see Section 2.5.1, “Editing the LDAP Configuration File”.

2.8. Support for Multi-Domain

The ACME LDAP agent can be configured to login from different domains. This is helpful in a scenario where users from different locations or departments in an organization can login by prefixing domain to username.

Note

The domain name is not case sensitive, must not contain any special characters, and must not be greater than 25 characters in length. If a domain name is not specified at the username prompt then the user will be authenticated against the default domain (the domain name specified in the first configuration file mentioned in the logical name LDAPACME$INIT).

Example 3 – Multi-Domain Users and Session Details

```
"AMERICAS\bwills"
"EMEA\John Doe"
"ASIAPACIFIC\Shaun Marsh"

Username: "Americas\bwills"
Password:
   VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1
**** Logon authenticated by LDAP ****
    OpenVMS password has been synchronized with external password
$ SH PROC

 4-NOV-2013 21:04:27.26   User: BWILLS           Process ID:   2020026F
                          Node: BENZ             Process name: "BWILLS"

Terminal:           TNA14:  (Host: LOCALHOST Locn:  FTA5:/SYSTEM)
User Identifier:    [BWILLS]
Base priority:      4
Default file spec:  SYS$SYSDEVICE:[BWILLS]
Number of Kthreads: 1 (System-wide limit: 2)

Username: "EMEA\John Doe"
Password:
  VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1
  on node BENZ Last interactive login on Monday, 30-NOV-2015 00:21:23.54
**** Logon authenticated by LDAP ****
$ SH PROC

 4-NOV-2013 21:17:08.99   User: JDOE             Process ID:   20200270
                          Node: BENZ             Process name: "JDOE"

Terminal:           TNA15:  (Host: LOCALHOST Locn: _TNA14:/BWILLS)
User Identifier:    [JDOE]
Base priority:      4
Default file spec:  SYS$SYSDEVICE:[JDOE]
Number of Kthreads: 1 (System-wide limit: 2)

Devices allocated:  TNA15:

Username: "Asiapacific\Shaun Marsh"
Password:
   VMS Software, Inc. OpenVMS(TM) IA64 Operating System, Version V8.4-1H1
   on node BENZ Last interactive login on Monday, 30-Nov-2015 10:40:09.01
**** Logon authenticated by LDAP ****
$ SH PROC

 4-NOV-2013 21:31:26.80   User: SMARSH           Process ID:   20200271
                          Node: BENZ             Process name: "SMARSH"

Terminal:           TNA16: (Host: LOCALHOST Locn: _TNA15:/JDOE)
User Identifier:    [SMARSH]
Base priority:      4
Default file spec:  SYS$SYSDEVICE:[SMARSH]
Number of Kthreads: 1 (System-wide limit: 2)

Devices allocated:  TNA16:
```

Chapter 3. Global and Local Mapping

The authentication method for the ACME LDAP agent on previous versions of OpenVMS only supported one-to-one mapping for users. In one-to-one mapping, the user logging in to an OpenVMS system from an LDAP server must have a matching username in the SYSUAF.DAT file. Hence, a user must login with the exact username entry stored in the SYSUAF.DAT file.

To overcome this limitation of one-to-one mapping, the ACME LDAP agent uses the concept of global and local mapping. The following diagrams explain the limitations of one-to-one mapping and show how global or local mapping overcomes the limitations. In this section, "jdoe" is used as a sample account in the SYSUAF.DAT file and "John Doe" as the sample domain user name.

Figure 3.1. One-to-One Mapping

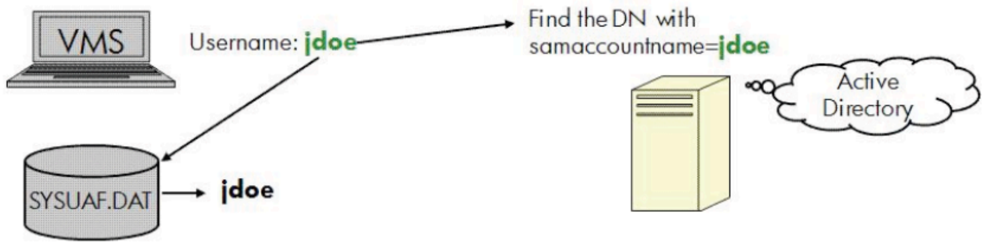


Figure 3.2. One-to-One Mapping Issue

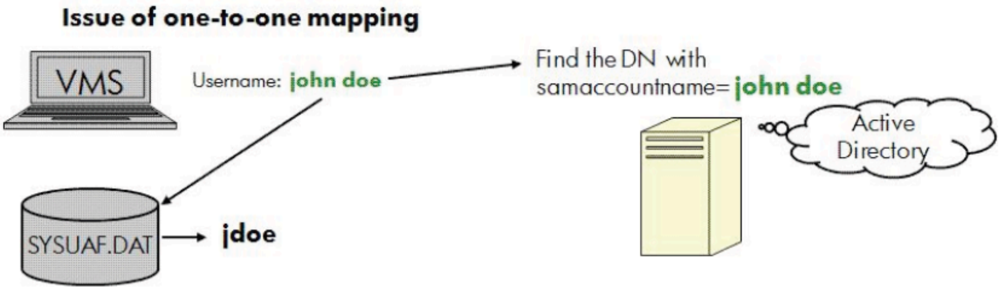


Figure 3.2, “One-to-One Mapping Issue” illustrates that in one-to-one mapping, the system is not able to match the username `John Doe` with the username in the `SYSUAF.DAT`, where it is stored as `jdoe`.

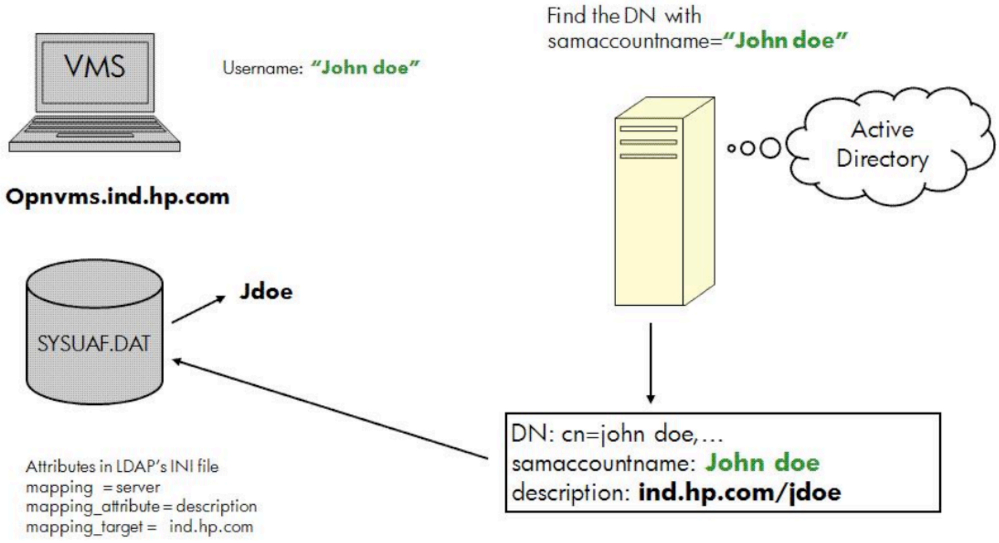
Using the global and local mapping:

- Users can enter the user name that is common across the domain at the user name prompt of OpenVMS.
- The username is mapped to a different name in the SYSUAF.DAT file during login.
- After login, the OpenVMS session uses the name and the privileges specified in the SYSUAF.DAT for all purposes.
- The **`SET PASSWORD`** command has the capability to understand that this is a mapped user and synchronize any password change to the directory server.

In global mapping, the user's login name is mapped based on some attributes stored in the directory server. In local mapping, a text database file is used to store the LDAP user name (name of the user in the domain) and the name in the SYSUAF.DAT in the .CSV format.

The following two figures illustrate global mapping and local mapping, respectively:

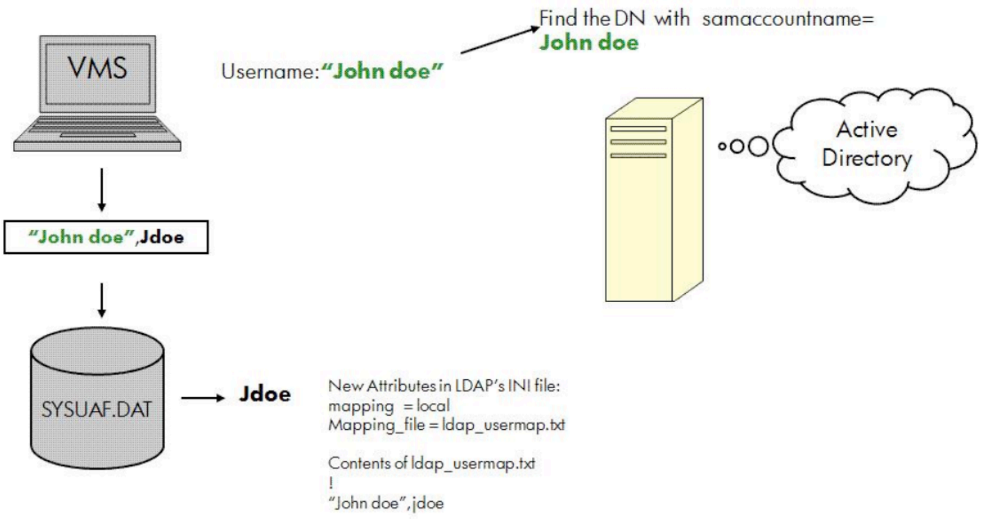
Figure 3.3. Global Mapping



In Figure 3.3, “Global Mapping”, the username `John Doe` is mapped with `jdoe` in the `SYSUAF.DAT` and `John Doe` in the `Active Directory`.

Three new directives (`mapping`, `mapping_attribute`, and `mapping _target`) are added to configure global mapping. For more information on global mapping directives, see Table 2.1, “LDAP Configuration Attributes”.

Figure 3.4. Local Mapping


VMS
Username:"John doe"
Find the DN with samaccountname=
John doe
Active Directory
"John doe",Jdoe
SYSUAF.DAT
Jdoe
New Attributes in LDAP's INI file:
mapping = local
Mapping_file = ldap_usermap.txt
Contents of ldap_usermap.txt
!
"John doe",jdoe

In Figure 3.4, “Local Mapping”, the username `John Doe` is mapped with `jdoe` and `John Doe` in the local database file.

Two new directives (`mapping` and `mapping_file`) are added to configure local mapping. For more information on local mapping directives, see Table 2.1, “LDAP Configuration Attributes”.

3.1. Configuring Global and Local Mapping

This section provides a user scenario on how to configure global mapping and local mapping with an OpenVMS ACME LDAP agent.

Global Mapping Configuration

In the SYSUAF.DAT file, the username is stored as "jdoe" and "jhardy". To enable global mapping, perform the following steps:

1. Update the attributes in SYS$STARTUP:LDAPACME$CONFIG-STD.INI file along with the other mandatory attributes:

   ```
   mapping = server
   mapping_attribute = description
   mapping_target = VMSusers.vmssoftware.com
   ```

   For example, two users (John Doe and Joe Hardy) have the following attributes specified in the user profile of the Active directory:

   ```
   DN: cn=john doe,…
   samaccountname: John Doe
   description: VMSUsers.vmssoftware.com/jdoe
   DN: cn=jhardy,…
   samaccountname: jhardy
   description: VMSUsers.vmssoftware.com/jhardy
   ```

2. Restart the ACME server using the following commands:

   ```
   $ SET SERVER ACME/EXIT/WAIT
   $ SET SERVER ACME/START=AUTO
   ```

3. Login to the host system using the login "John Doe" for the user "John Doe".

   Note

   At the user name prompt you must give this name in quotes, as the name has a space (special character) in-between.

4. Login to the host system using the login "jhardy" for the other user.

Local Mapping Configuration

To enable local mapping, perform the following steps:

1. Make a copy of the SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT _TEMPLATE and rename it to a filename of your choice. For example, SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT on the OpenVMS system.

2. Update the SYS$STARTUP:LDAP_LOCALUSER_DATABASE.TXT with the LDAP username and VMS username separated by a comma. If the LDAP username contains spaces, commas, or exclamation points, provide it within quotes:

```
"John Doe",jdoe
jhardy,jhardy
```

For example, two users John Doe and Joe Hardy have the following attributes specified in the user profile of the Active directory:

```
DN: cn=john doe,…
samaccountname: John Doe
DN: cn=jhardy,…
samaccountname: jhardy
```

3. Update the directives in the SYS$STARTUP:LDAPACME$CONFIG-STD.INI file along with the other mandatory attributes:

```
mapping = local
mapping_file = SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE.TXT
```

4. Load the new database file by either of the following methods.

   - Restart the ACME server using the following commands:

   ```
   $ SET SERVER ACME/EXIT/WAIT
   $ SET SERVER ACME/START=AUTO
   ```

   - Use LDAP_LOAD_LOCALUSER_DATABASE.EXE:

   ```
   $ load_localuser_db:=="$SYS$SYSTEM:LDAP_LOAD_LOCALUSER_DATABASE.EXE"
   $ load_localuser_db SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE.TXT
   ```

   In cases of multi-domain support, the procedure to load the local user database is as follows (here the tool takes one extra argument, "domain"):

   ```
   $load_localuser_db:=="$SYS$SYSTEM:LDAP_LOAD_LOCALUSER_DATABASE.EXE"
   $load_localuser_db SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE_AMERICAS.TXT AMERICAS
   $load_localuser_db SYS$COMMON:[SYS$STARTUP]LDAP_LOCALUSER_DATABASE_EMEA.TXT EMEA
   ```

5. Login to the host system using the login "John Doe" and jhardy.

Chapter 4. User Scenario of Configuring an OpenVMS ACME LDAP Agent

This chapter provides a user scenario on how to configure an OpenVMS ACME LDAP agent. It also provides the steps to extract the relevant values from the Active directory server to populate the ACME LDAP configuration file.

Note

This chapter aims at providing the end-user with a detailed overview of configuring an OpenVMS ACME LDAP agent.

Sample account names such as, "query_account" have been used throughout this chapter and must not be considered as a standard proxy account name. You can create any account of your choice. Similarly, other accounts and system names used in this chapter are also examples and you can use any account name or system of your choice.

Figure 4.1. ACME LDAP Process Flow Diagram

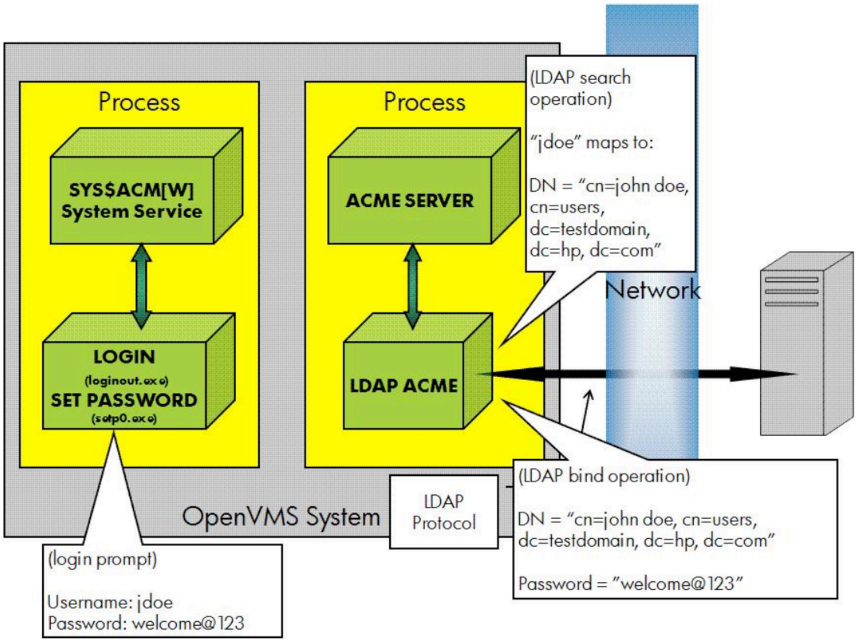


Figure 4.1, "ACME LDAP Process Flow Diagram" illustrates how a VMS user logs in to a VMS system using LDAP authentication. In this figure, two systems are involved, which communicate over TCP/IP.

The gray box on the left is the VMS system with enhanced versions of `LOGINOUT.EXE` and `SETP0.EXE` installed and the ACME LDAP agent running within the ACME_SERVER process.

On the right is the Active directory server running Windows Server 2003. Active Directory is also an LDAP server.

The ACME LDAP agent communicates with Active directory using LDAP protocol over a TCP session, which can be protected by SSL (required for Active directory LDAP password changes). The LDAP "search" and "bind" operations are standard LDAP operations accessed through standard C bindings. These are operations that are supported with any standard LDAP server and are used pervasively in many applications to provide LDAP-based authentication services.

4.1. Extracting ACME LDAP Configuration Parameter Values

You require the following information from the Active directory to populate the LDAP INI configuration file:

- LDAP port (this is usually 389 for the non-secure port and 636 for the secure port). For detailed steps on how to obtain this information, see Section 4.1.1, “Querying LDAP Port”.
- Base Distinguished Name (DN) under which all users are present.
- Distinguished Name and password of the "query_account".
- Login attribute (usually "samaccountname").

The base distinguished name (`base_dn` directive), the distinguished name of the query_account (`bind_dn` directive), and the samaccountname (`login_attribute` directive) are obtained from the database log file, .ldf file. For more information on how to obtain the specific attribute value, see Section 4.1.2, “Extracting base_dn, bind_dn, and login_attribute”.

4.1.1. Querying LDAP Port

To query LDAP ports, you can install the PortQryUI tool provided by Microsoft. This tool is available for download from the Microsoft website ().

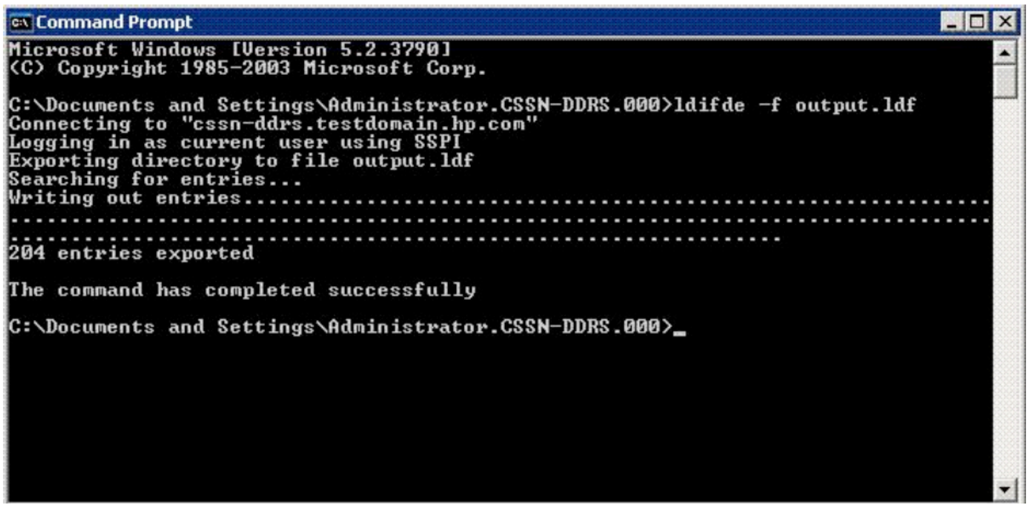
You can use any other query tool of your choice.

4.1.2. Extracting base_dn, bind_dn, and login_attribute

You can extract the values for the `base_dn`, `bind_dn`, and `login_attribute` directives (in the ACME LDAP configuration file) from the .ldf file.

To extract the .ldf file, at the command prompt, enter the following command on your Windows system:

```
ldifde -f <filename>.ldf
```

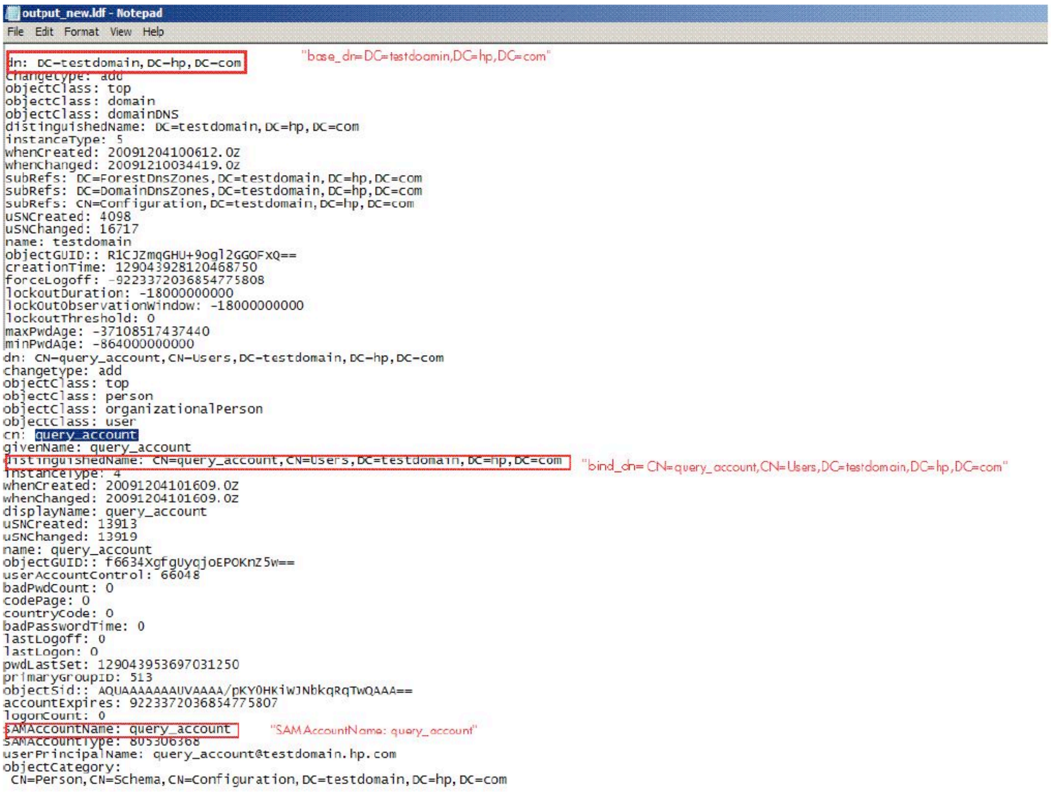


```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator.CSSN-DDRS.000>ldifde -f output.ldf
Connecting to "cssn-ddrs.testdomain.hp.com"
Logging in as current user using SSPI
Exporting directory to file output.ldf
Searching for entries...
Writing out entries.............................................................
................................................................................
...............................................................
204 entries exported

The command has completed successfully

C:\Documents and Settings\Administrator.CSSN-DDRS.000>_
```

After the .ldf file is extracted, copy the `base_dn` and `bind_dn` values. For more information on the `base_dn` and `bind_dn` directives, see Table 2.1, “LDAP Configuration Attributes”.

Figure 4.2, “Sample LDF File” shows a sample .ldf file. Here, the account, `query_account` is identified as the binding account. The `base_dn` and `bind_dn` values are highlighted.

Figure 4.2. Sample LDF File



output_new.ldf - Notepad
File Edit Format View Help
"base_dn=DC=testdoamin,DC=hp,DC=com"
"bind_dn= CN=query_account,CN=Users,DC=testdomain,DC=hp,DC=com"
"SAM AccountName: query_account"

```
dn: DC=testdomain,DC=hp,DC=com
changetype: add
objectClass: top
objectClass: domain
objectClass: domainDNS
distinguishedName: DC=testdomain,DC=hp,DC=com
instanceType: 5
whenCreated: 20091204100612.0Z
whenChanged: 20091210034419.0Z
subRefs: DC=ForestDnsZones,DC=testdomain,DC=hp,DC=com
subRefs: DC=DomainDnsZones,DC=testdomain,DC=hp,DC=com
subRefs: CN=Configuration,DC=testdomain,DC=hp,DC=com
uSNCreated: 4098
uSNChanged: 16717
name: testdomain
objectGUID:: R1CJZmqGHU+9ogl2GGOFXQ==
creationTime: 129043928120468750
forceLogoff: -9223372036854775808
lockoutDuration: -18000000000
lockOutObservationWindow: -18000000000
lockoutThreshold: 0
maxPwdAge: -37108517437440
minPwdAge: -864000000000
dn: CN=query_account,CN=Users,DC=testdomain,DC=hp,DC=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: query_account
givenName: query_account
distinguishedName: CN=query_account,CN=Users,DC=testdomain,DC=hp,DC=com
instanceType: 4
whenCreated: 20091204101609.0Z
whenChanged: 20091204101609.0Z
displayName: query_account
uSNCreated: 13913
uSNChanged: 13919
name: query_account
objectGUID:: f6634XgfgUyqjoEPOKnZ5w==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 0
lastLogoff: 0
lastLogon: 0
pwdLastSet: 129043953697031250
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAA/pKY0HKiWJNbkqRqTwQAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: query_account
sAMAccountType: 805306368
userPrincipalName: query_account@testdomain.hp.com
objectCategory:
 CN=Person,CN=Schema,CN=Configuration,DC=testdomain,DC=hp,DC=com
```

4.2. Configuring an ACME LDAP Agent for Non-Secure Ports

To configure an ACME LDAP agent on a non-secure port, do the following:

1. Enable the SYS$ACM enabled login and ACME LDAP as explained in Section 2.3, “Enabling the SYS$ACM (ACMELOGIN) Enabled LOGIN”.

2. Check whether the images are loaded correctly:

   ```
   ANALYZE/IMAGE/INTER SYS$COMMON:[SYSEXE]LOGINOUT.EXE
   ```

   ```
   $ ANALYZE/IMAGE/INTER SYS$COMMON:[SYSEXE]LOGINOUT.EXE
   This is an OpenVMS IA64 (Elf format) executable image file

   Image Identification Information, in section 3.

       Image name:                                 "LOGIN_ACME"
       Global Symbol Table name:                   "LOGIN_ACME"
       Image file identification:                  "LOGIN_ACME"
       Image build identification:                 "XFW1-C6E-000000"
       Link identification:                        "Linker I02-37"
       Link Date/Time:                              9-MAR-2021 22:45:26.17
   ```

   ```
   ANALYZE/IMAGE/INTER SYS$COMMON:[SYSEXE]SETP0.EXE
   ```

   ```
   $ ANALYZE/IMAGE/INTER SYS$COMMON:[SYSEXE]SETP0.EXE
   This is an OpenVMS IA64 (Elf format) executable image file

   Image Identification Information, in section 3.

       Image name:                                 "SETP0_ACME"
       Global Symbol Table name:                   "SETP0_ACME"
       Image file identification:                  "LOGIN_ACME"
       Image build identification:                 "XFW1-C6E-000000"
       Link identification:                        "Linker I02-37"
       Link Date/Time:                              9-MAR-2021 22:46:01.27
   ```

3. Set up the LDAP persona extension. For more information on how to set the persona extension, see Section 2.4, “Setting Up the LDAP Persona Extension”.

4. Restart the OpenVMS system after setting the persona extension.

5. For a non-secure port, enter the following values for the attributes in the LDAP configuration file, SYS$STARTUP:LDAPACME$CONFIG-STD.INI:

   server = cssn-ddrs.testdomain.vmssoftware.com

   > Ensure that you are able to make a **`TCPIP PING cssn-ddrs.testdomain.vmssoftware.com`** to the Active directory system port = 389. This is the default value for a non-secure port.

   bind_dn = CN=query_account,CN=Users,DC=testdomain,DC=vmssoftware,DC=com

   > This value can be obtained from the .ldf file. For information on how to extract the value, see Section 4.1.2, “Extracting base_dn, bind_dn, and login_attribute”.

   bind_password = welcome@123

   > This is the password given for the query_account in the Active directory.

base_dn = DC=testdomain,DC=vmssoftware,DC=com

This is the base account under which all other accounts reside.

login_attribute = samaccountname

scope = sub

Retain the default value "sub".

port_security = none

Since this is a non-secure port, replace the default value with "none".

password_type = active-directory

Replace the default value with active-directory since the configuration is done with an Active directory.

The populated configuration file will be as shown:

```
server = cssn-ddrs.testdomain.vmssoftware.com
port = 389
bind_dn = CN=query_account,CN=Users,DC=testdomain,DC=vmssoftware,DC=com
bind_password = welcome@123
base_dn = DC=testdomain,DC=vmssoftware,DC=com
login_attribute = samaccountname
scope = sub
port_security = none
password_type = active-directory
```

6. Add the following logical to the SYS$MANAGER:ACME$START.COM:

   ```
   $ DEFINE/SYSTEM/EXECUTIVE LDAPACME$INIT -
   _$SYS$STARTUP:LDAPACME$CONFIG-STD.INI
   ```

   Additionally, uncomment the @SYS$STARTUP:LDAPACME$STARTUP-STD.

7. Restart the ACME server using the following commands:

   ```
   $ SET SERVER ACME/EXIT/WAIT
   $ SET SERVER ACME/START=AUTO
   ```

8. Execute the **`SHOW SERVER ACME/FULL`** command to check if the ACME LDAP agent has been loaded:

   ```
   $ SHOW SERVER ACME/FULL
   ACME Information on node EARWIG  18-FEB-2010 06:03:42.00  Uptime 0
    00:15:24

   ACME Server id: 2  State: Processing New Requests
      Agents Loaded:        2   Active:       2
      Thread Maximum:       1   Count:        1
      Request Maximum:    826   Count:        0
      Requests awaiting service:              0
      Requests awaiting dialogue:             0
   ```

```
      Requests awaiting AST:                 0
      Requests awaiting resource:            0

   ACME Agent id: 1  State: Active
      Name: "VMS"
      Image: "DISK$I64SYS:[VMS$COMMON.SYSLIB]VMS$VMS_ACMESHR.EXE;1"
      Identification: "VMS ACME built  20-SEP-2006"
      Information: "No requests completed since the last startup"
      Domain of Interpretation: Yes
      Execution Order:       1
      Credentials Type:      1   Name: "VMS"
   Resource wait count:                      0

   ACME Agent id: 2 State: Active
      Name: "LDAP-STD"
      Image: "DISK$I64SYS:[VMS$COMMON.SYSLIB]LDAPACME$LDAP-
   STD_ACMESHR.EXE;1"
      Identification: "ACME LDAP Standard V1.5"
      Information: "ACME_LDAP_DOI Agent is initialized"
      Domain of Interpretation: Yes
      Execution Order:       2
      Credentials Type:      3   Name: "LDAP"
      Resource wait count:                   0
```

9. Add the user "jdoe" to the SYSUAF.DAT file:

```
@SYS$COMMON:[SYSHLP.EXAMPLES]ADDUSER.COM
**********************************************************************
*  Creating a NEW user account... If at ANY TIME you need help about a   *
*  prompt, just type "?".                                                 *
**********************************************************************

Username(s) - separate by commas: jdoe

*** Processing JDOE's account ***

Full name for JDOE: John Doe
Password (password is not echoed to terminal) [JDOE]:

UIC Group number [200]:
UIC Member number: 201
Account name: TEST
Privileges [TMPMBX,NETMBX]:

Login directory [JDOE]:
Login device [SYS$SYSDEVICE:]:

%CREATE-I-EXISTS, SYS$SYSDEVICE:[JDOE] already exists %UAF-
IPWDLESSMIN, new password is shorter than minimum password length %UAF-
E-UAEERR, invalid user name, user name already exists %UAF-I-NOMODS,
no modifications made to system authorization file %UAF-I-RDBNOMODS,
no modifications made to rights database

Check newly created account:

Username: JDOE                           Owner:
Account:  TEST                           UIC:     [201,2011] ([JDOE])
CLI:      DCL                            Tables: DCLTABLES
Default:  SYS$SYSDEVICE:[JDOE]
LGICMD:
Flags:  VMSAuth
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                     Sat Sun
```

```
No access restrictions
Expiration:            (none)     Pwdminimum:  6   Login Fails:     1
Pwdlifetime:         90 00:00     Pwdchange:      (pre-expired)
Last Login:            (none) (interactive),            (none) (non-interactive)
Maxjobs:          0 Fillm:       128  Bytlm:        128000
Maxacctjobs:      0 Shrfillm:      0  Pbytlm:            0
Maxdetach:        0 BIOlm:       150  JTquota:        4096
Prclm:            8 DIOlm:       150  WSdef:          4096
Prio:             4 ASTlm:       300  WSquo:          8192
Queprio:          4 TQElm:       100  WSextent:      16384
CPU:         (none) Enqlm:      4000  Pgflquo:      256000
Authorized Privileges:
  NETMBX       TMPMBX
Default Privileges:
  NETMBX       TMPMBX
%UAF-I-NOMODS, no modifications made to system authorization
file %UAF-I-RDBNOMODS, no modifications made to rights database

Is everything satisfactory with the account [YES]:
```

10. Set ExtAuth and VMSAuth flag for the user "jdoe" (for information about adding a SYSUAF account, see Section 2.6, “Specifying EXTAUTH and VMSAUTH Flags on OpenVMS”):

```
$ SET DEF SYS$SYSTEM
$ MC AUTHORIZE
UAF> modify jdoe/flags=(EXTAUTH,VMSAUTH)
%UAF-I-MDFYMSG, user record(s) updated
UAF> SHOW jdoe

Username: JDOE                             Owner:
Account:  TEST                             UIC:     [201,2011] ([JDOE])
CLI:      DCL                              Tables: DCLTABLES
Default:  SYS$SYSDEVICE:[JDOE]
LGICMD:
Flags:  VMSAuth
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                     Sat Sun
No access restrictions
Expiration:            (none)     Pwdminimum:  6   Login Fails:     1
Pwdlifetime:         90 00:00     Pwdchange:      (pre-expired)
Last Login:            (none) (interactive),            (none) (non-interactive)
Maxjobs:          0 Fillm:       128  Bytlm:        128000
Maxacctjobs:      0 Shrfillm:      0  Pbytlm:            0
Maxdetach:        0 BIOlm:       150  JTquota:        4096
Prclm:            8 DIOlm:       150  WSdef:          4096
Prio:             4 ASTlm:       300  WSquo:          8192
Queprio:          4 TQElm:       100  WSextent:      16384
CPU:         (none) Enqlm:      4000  Pgflquo:      256000
Authorized Privileges:
  NETMBX       TMPMBX
Default Privileges:
  NETMBX       TMPMBX
UAF>
```

11. Login to the system as user "jdoe".

4.3. Configuring an ACME LDAP Agent for Secure Ports

To configure an ACME LDAP agent on a secure port, do the following:

1. Update the LDAP configuration file SYS$STARTUP:LDAPACME$CONFIG-STD.INI similar to how the file was updated in Section 4.2, “Configuring an ACME LDAP Agent for Non-Secure Ports”. The only difference is the values provided to the `port` and `port_security` directives.

   See the following sample configuration files:

   ```
   server = cssn- ddrs.testdomain.vmssoftware.com
   port = 636
   bind_dn = CN=query_account,CN=Users,DC=testdomain,DC=vmssoftware,DC=com
   bind_password = welcome@123
   base_dn = DC=testdomain,DC=vmssoftware,DC=com
   scope = sub
   port_security = ssl
   password_type = active- directory
   ```

   OR

   ```
   server = cssn- ddrs.testdomain.vmssoftware.com
   port = 389
   bind_dn = CN=query_account,CN=Users,DC=testdomain,DC=vmssoftware,DC=com
   bind_password = welcome@123
   base_dn = DC=testdomain,DC=vmssoftware,DC=com
   scope = sub
   port_security = starttls
   password_type = active-directory
   ```

2. Restart the ACME_SERVER and check the login as explained in Section 4.2, “Configuring an ACME LDAP Agent for Non-Secure Ports”.

4.4. Configuring ACME LDAP to Utilize Multi-Domain Support

Perform the following steps to configure ACME LDAP to utilize the multi-domain support:

1. Identify the different domains in the organization, and the respective directory servers which hosts these domains.

   For example:

   ```
   Domain: Americas     Server: Boston.americas.vmssoftware.com
   Domain: EMEA         Server: London.EMEA.vmssoftware.com
   Domain: Asiapacific  Server: Sydney.asiapacific.vmssoftware.com
   ```

2. Create a separate INI file for each domain.

   For example:

   ```
   SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_Americas.INI
   SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_EMEA.INI
   SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_Asiapacific.INI
   ```

3. Enter the required attributes value in each INI file, mentioning the attribute domain and the respective domain name as shown in the following example.

   Examples of INI files after configuring the ACME LDAP agent for multi-domain support are as follows:

-
  ```
  SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_Americas.INI
  file server = BOSTON.AMERICAS.VMSSOFTWARE.COM
  port = 636
  bind_dn = CN=query_account,CN=Users,DC=AMERICAS,DC=vmssoftware,DC=com
  bind_password = welcome@123
  base_dn = DC=AMERICAS,DC=vmssoftware,DC=com
  login_attribute = sAMAccountName
  scope = sub
  port_security = ssl
  password_type = active- directory
  domain = AMERICAS
  ```

-
  ```
  SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_EMEA.INI
  file server = LONDON.EMEA.VMSSOFTWARE.COM
  port = 389
  bind_dn = CN=query_account,CN=Users,DC=EMEA,DC=vmssoftware,DC=com
  bind_password = welcome@123
  base_dn = DC=EMEA,DC=vmssoftware,DC=com
  login_attribute = sAMAccountName
  scope = sub
  port_security = none
  password_type = active- directory
  domain=EMEA
  ```

-
  ```
  SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_Asiapacific.INI
  file server = SYDNEY.ASIAPACIFIC.VMSSOFTWARE.COM
  port = 636
  bind_dn = CN=query_account,CN=Users,DC=ASIAPACIFIC,DC=vmssoftware,DC=com
  bind_password = welcome@123
  base_dn = DC=ASIAPACIFIC,DC=vmssoftware,DC=com
  login_attribute = sAMAccountName
  scope = sub
  port_security = ssl
  password_type = active- directory
  domain = ASIAPACIFIC
  ```

4. After completing the aforementioned steps, edit SYS$MANAGER:ACME$START.COM, and define a multilevel system wide logical LDAPACME$INIT to point all the domain's INI files using comma separated values.

   For example:

   ```
   $ DEFINE/SYSTEM/EXECUTIVE LDAPACME$INIT -
   _$SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_Americas.INI,-
   _$SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_EMEA.INI,-
   _$SYS$COMMON:[SYS$STARTUP]LDAPACME$CONFIG-STD_Asiapacific.INI
   ```

5. Restart the ACME server using the following commands:

   ```
   $ SET SERVER ACME/EXIT/WAIT
   $ SET SERVER ACME/START=AUTO
   ```

Chapter 5. Troubleshooting

Problem

System displays the following error when @SYS$STARTUP:ACME$START.COM is executed:

```
$ @SYS$STARTUP:ACME$START.COM
Please ensure the following logical is defined
  /SYSTEM/EXECUTIVE_MODE LDAPACME$INIT
```

Solution

The LDAPACME$INIT logical is not defined before the
@SYS$STARTUP:LDAPACME$STARTUP- STD command in
SYS$COMMON:[SYSMGR]ACME$START.COM. For more information, see the steps in
Section 2.5.1, “Editing the LDAP Configuration File”.

Problem

When @SYS$STARTUP:ACME$START.COM is executed, the system displays the following error, all ACME agents are in stopped state when using the **SHOW SERVER ACME/FULL** command, and new logins are not permitted:

```
$ @SYS$STARTUP:ACME$START.COM
%ACME-E-INVPARAMETER, parameter selector or descriptor is invalid
```

Solution

The LDAPACME$INIT logical is defined to a wrong INI file name. Perform the following steps:

1. Deassign the LDAPACME$INIT logical:

   ```
   $ DEASSIGN /SYSTEM/EXEC LDAPACME$INIT
   ```

2. Stop the ACME Server process:

   ```
   $ SET SERVER ACME/EXIT/WAIT
   ```

3. Correct the LDAPACME$INIT logical to point to the right path inside SYS$STARTUP:ACME$START.COM.

4. Start the ACME server in auto mode so that it starts the ACME LDAP agent during startup:

   ```
   $ SET SERVER ACME/START=AUTO
   ```

Problem

The **SHOW SERVER ACME/FULL** command does not display the LDAP agent.

```
$ SHOW SERVER ACME/FULL
ACME Information on node EARWIG  18-FEB-2010 05:50:06.40  Uptime 0 00:01:48
```

```
ACME Server id: 2   State: Processing New Requests
   Agents Loaded:          1  Active:       1
   Thread Maximum:         1  Count:        1
   Request Maximum:      826  Count:        0
   Requests awaiting service:               0
   Requests awaiting dialogue:              0
   Requests awaiting AST:                   0
   Requests awaiting resource:              0
   Logging status: Active
   Tracing status: Inactive
   Log file: "SYS$SYSROOT:[SYSMGR]ACME$SERVER.LOG;17"

ACME Agent id: 1    State: Active
   Name: "VMS"
   Image: "DISK$I64SYS:[VMS$COMMON.SYSLIB]VMS$VMS_ACMESHR.EXE;1"
   Identification: "VMS ACME built  20-SEP-2006"
   Information: "No requests completed since the last startup"
   Domain of Interpretation: Yes
   Execution Order:       1
   Credentials Type:      1  Name: "VMS"
   Resource wait count:                     0
$
```

Solution

Check if the SYS$STARTUP:ACME$START.COM has been updated with the LDAP logical names and @SYS$STARTUP:LDAPACME$STARTUP-STD ! LDAP command is uncommented in the file. For more information on updating the SYS$STARTUP:ACME$START.COM, see Section 2.5.1, “Editing the LDAP Configuration File”.

```
ACME Server id: 2  State: Processing New Requests
   Agents Loaded:         2  Active:       2
   Thread Maximum:        1  Count:        1
   Request Maximum:     826  Count:        0
   Requests awaiting service:              0
   Requests awaiting dialogue:             0
   Requests awaiting AST:                  0
   Requests awaiting resource:             0
   Logging status: Active
   Tracing status: Inactive
   Log file: "SYS$SYSROOT:[SYSMGR]ACME$SERVER.LOG;19"

ACME Agent id:  1 State: Active
   Name: "VMS"
   Image: "DISK$I64SYS:[VMS$COMMON.SYSLIB]VMS$VMS_ACMESHR.EXE;1"
   Identification: "VMS ACME built   20-SEP-2006"
   Information: "No requests completed since the last startup"
   Domain of Interpretation: Yes
   Execution Order:       1
   Credentials Type:      1  Name: "VMS"
   Resource wait count:                    0

ACME Agent id: 2  State: Active
   Name: "LDAP-STD"
   Image: "DISK$I64SYS:[VMS$COMMON.SYSLIB]LDAPACME$LDAP-STD_ACMESHR.EXE;1"
   Identification: "LDAP ACME Standard V1.5"
   Information: "ACME_LDAP_DOI Agent is initialized"
   Domain of Interpretation: Yes
   Execution Order:       2
   Credentials Type:      3  Name: "LDAP"
   Resource wait count:                    0
$
```

Problem

The ACME LDAP configuration is correct, but the user is unable to log in.

Solution 1

Use the **PING** command to check whether the LDAP server provided in the `server` directive of the LDAP INI file is reachable:

```
$ TCPIP PING
PING earwig (15.146.235.235): 56 data bytes
64 bytes from 15.146.235.235: icmp_seq=0 ttl=64 time=0 ms
64 bytes from 15.146.235.235: icmp_seq=1 ttl=64 time=0 ms
64 bytes from 15.146.235.235: icmp_seq=2 ttl=64 time=0 ms
64 bytes from 15.146.235.235: icmp_seq=3 ttl=64 time=0 ms

----earwig PING Statistics----
4 packets transmitted, 4 packets received, 0% packet
loss round-trip (ms) min/avg/max = 0/0/0 ms
```

Solution 2

Ensure that the ExtAuth flag is provided for the user in the SYSUAF.DAT file.

Solution 3

Use **TCPDUMP** to check whether data is flowing on the configured LDAP port.

```
$ TCPDUMP -W TCPDUMP.ENC TCP PORT 389
tcpdump: Filtering in user process
tcpdump: listening on WE1, link-type EN10MB (Ethernet), capture size 96
bytes *CANCEL*

24 packets captured
24 packets received by filter
0 packets dropped by
kernel $ dir .enc

Directory SYS$SYSROOT:[SYSMGR]

TCPDUMP.ENC;1

Total of 1 file.
$
$ TCPDUMP -R TCPDUMP.ENC
reading from file tcpdump.enc, link-type EN10MB (Ethernet)
05:39:16.726000 IP opnvms.ind.vmssoftware.com.49160 > CSSN-DDRS.TESTDOMAIN.VMSSOFTWARE.COM.389: S
1252791091:1252791091(0) win 61440 <mss 1460,nop,wscale 0>
05:39:16.726000 IP CSSN-DDRS.TESTDOMAIN.VMSSOFTWARE.COM.389 > opnvms.ind.vmssoftware.com.49160: S
1725693481:1725693481(0) ack 1252791092 win 16384 <mss 1460,nop,wscale 0>
05:39:16.726000 IP opnvms.ind.vmssoftware.com.49160 > CSSN-DDRS.TESTDOMAIN.VMSSOFTWARE.COM.389: .
 ack 1 win 62780
05:39:16.726000 IP opnvms.ind.vmssoftware.com.49160 > CSSN-DDRS.TESTDOMAIN.VMSSOFTWARE.COM.389: P
 1:78(77) ack 1
```

```
win 62780 05:39:16.728000 IP CSSN-DDRS.TESTDOMAIN.VMSSOFTWARE.COM.389 >
 opnvms.ind.vmssoftware.com.49160: P
1:23(22) ack 78 win 65458 05:39:16.729000 IP opnvms.ind.vmssoftware.com.49160 > CSSN -
DDRS.TESTDOMAIN.VMSSOFTWARE.COM.389: P 78:154(76) ack 23 win 62780
```

Solution 4 (Requires C Compiler)

To troubleshoot issues with the LDAP configuration, use a compiled version of SYS$EXAMPLES:LDAP_EXAMPLE.C.

Once compiled, the LDAP_EXAMPLE.EXE file can be used to search the directory server. The LDAP_EXAMPLE.EXE file accepts arguments similar to the directives in the LDAP INI configuration file. As a result, you can populate the INI file with the correct directive information, based on the output of LDAP_EXAMPLE.EXE.

```
$ SET DEF SYS$EXAMPLES
$ CC LDAP_EXAMPLE
$ LINK LDAP_EXAMPLE $ LDAP_EXAMPLE:=="$SYS$EXAMPLES:LDAP_EXAMPLE.EXE"
$ LDAP_EXAMPLE
$ LDAP_EXAMPLE
Usage:ldap_example server port bind_dn bind_password port_security cafile base_dn filter

[attributes] Mandatory arguments : For specifying NULL values use ""

server        --> The node which is providing LDAP access to a directory
port          --> The port through which to search
bind_dn       --> The bind dn, enclose in double quotes. Specify a "" if
                  anonymous bind is supported by LDAP directory server.
bind_password --> The bind password. Specify a "" if anonymous bind
                  is supported by LDAP directory server.
port_security --> The port security "SSL" or "TLS". Specify a "" if
                  you are not using any port security.
cafile        --> The location of the ca file. Specify a "" if ca file is
                  not present.
base_dn       --> The base object in the directory for the search operation.
                  This is a required argument.
filter        --> The search filter to be used. Specify a "" if the LDAP
                  search needs to be done without filters.

Optional arguments :
attributes    --> An optional list of one or more attributes to be returned
                  for each matching record. If no attributes are specified,
                  then all user attributes will be returned.

Example :

$ ldap_example server1 389 "" "" "" "" "ou=vms,o=testcom" ""
$ ldap_example server1 389 "cn=admin,ou=vms,o=testcom" "WELCOME123" "" ""
$ ldap_example server1 389 "cn=admin,ou=vms,o=testcom" "WELCOME123" "" ""
- "ou=vms,o=testcom" "" "DN"
$ ldap_example server1 389 "cn=admin,ou=vms,o=testcom" "WELCOME123" "" ""
- "ou=vms,o=testcom" "" "DN" "SN"
$ ldap_example server2 389 -
"CN=query_account,CN=Users,DC=testdomain,DC=testcom,DC=com" -
"welcome@123" "" "" "CN=Users,DC=testdomain,DC=testcom,DC=com" -
"" "samaccountname"
$ ldap_example server2 636 -
"CN=query_account,CN=Users,DC=testdomain,DC=testcom,DC=com" -
"welcome@123" "SSL" "" "CN=Users,DC=testdomain,DC=testcom,DC=com" -
"" "samaccountname"
$ ldap_example server2 389 "CN=query_account,CN=Users,DC=testdomain,DC=testcom,DC=com" -
"welcome@123" "starttls" "" "CN=Users,DC=testdomain,DC=testcom,DC=com" -
"" "samaccountname"
$ ldap_example server2 636 "CN=query_account,CN=Users,DC=testdomain,DC=testcom,DC=com" -
"welcome@123" "SSL" "SYS$SYSROOT:[SYSMGR]server2.CER" -
"CN=Users,DC=testdomain,DC=testcom,DC=com" "" "samaccountname"

Program terminating
```

5.1. Frequently Asked Questions

What events can be traced using the `SET SERVER ACME/TRACE=<value>` command, and how do we interpret the traces?

You can view critical errors logged by the agent in ACME$SERVER.LOG without setting the **`SET SERVER ACME/TRACE=<value>`**. See Table 5.1, “Trace Flags” for setting the appropriate values. For example:

When the ACME LDAP agent is configured to a directory server that is not reachable, the following error messages are displayed:

```
%ACME-I-LOGAGENT, agent initiated log event on 25-FEB-2010 10:41:06.43 ❶
-ACME-I-THREAD, thread: id = 4, type = EXECUTION ❷
-ACME-I-REQUEST, request information, id = 1, function = AUTHENTICATE_PRINCIPAL ❸
-ACME-I-CLIENT, client information, PID = 2020044C ❹
-ACME-I-AGENT, agent information, ACME id = 2, name = LDAP-STD ❺
-ACME-I-CALLOUT, active callout routine = acme$co_accept_principal ❻
-ACME-I-CALLBACK, active callback routine = acme$cb_send_logfile ❼
-ACME_-I-TRACE, message from LDAP ACME agent: Internal error. LDAP search operation failed ❽
```

❶ Time of Log.
❷ Thread ID of the ACME Server causing this error.
❸ Function code passed to SYS$ACM.
❹ Process ID of the client talking to the ACME Server.
❺ Agent handling this request.
❻ Authentication routine handling the request.
❼ Callback routine.
❽ Status returned by the ACME agent.

The following is another example of giving port_security = nonenone instead of port_security = none in the configuration file:

```
%ACME-I-LOGAGENT, agent initiated log event on 25-FEB-2010
10:42:39.41 -ACME-I-THREAD, thread: id = 1, type = CONTROL
-ACME-I-CONTROL, control information, operation = STARTUP -ACME-
I-AGENT, agent information, ACME id = 2, name = LDAP-STD -ACME-
I-CALLOUT, active callout routine = acme$co_agent_startup -ACME-
I-CALLBACK, active callback routine = acme$cb_send_logfile
-ACME_-I-TRACE, MESSAGE FROM LDAP ACME agent: Reading the config file (LDAPACME$INIT) failed ❶
```

❶ Error message.

The information starting from `%ACME-I-` to the next `%ACME-I-` marks one trace.

When you execute **`SET SERVER ACME/TRACE=<value>`**, tracing is enabled and logged to the SYS$MANAGER:ACME$SERVER.LOG file.

You must search for the "MESSAGE FROM LDAP ACME agent" line in the ACME$SERVER.LOG to locate status messages returned by the LDAP ACME agent.

For details about the various flags that can be enabled for tracing, execute the **`HELP SET SERVER ACME/TRACE`** command on an OpenVMS system.

The following table provides details about the trace flags.

Table 5.1. Trace Flags

Bitmask	Event	Description
0	agent	Enable agent tracing.

Bitmask	Event	Description
1	general	General (non-specific) tracing.
2	vm	Virtual memory operations. That is, trace the memory allocation and de-allocation of both the ACME_SERVER and the agent (if the agent uses the memory services provided by ACME_SERVER process). [1]
3	ast	AST processing. Traces ASTs that are triggered by agents to the ACME_SERVER.
4	wqe	*WQE* parameter that flows between the ACME_SERVER process and agent.
5	report	Agent status or attribute operations.
6	message	Messaging operations.
7	dialog	Dialogue operations.
8	resource	Agent resource operations. Agents can request for some specific resource locks from the ACME_SERVER process.
9	callout	Agent callout routine. Routines that are implemented by individual agents such as ACME LDAP, that are called by the ACME_SERVER.
10	callout_status	Agent callout return status.

[1] Tracing is not enabled if the agent uses is own or standard (malloc, calloc, free) memory management routines.

For example, the following command would enable tracing of "agent", "general", "report", "message", "dialog", "callout", and "callout_status":

```
$ SET SERVER ACME/TRACE=1763
```

Chapter 6. Restrictions

This section lists the restrictions associated with the ACME LDAP agent.

6.1. Username and Password Restrictions

- Password modifications are made to the standard userPassword attribute. The details of the configuration attributes are described in Chapter 2, *Enabling and Configuring the ACME LDAP Agent*. The ldap_modify "replace" or "remove-old/add-new" semantics for password modifications can be configured to support a variety of directory servers based on the user requirements.

  The following LDAP password policy client controls are supported to warn users of password expiration events:

  ```
  Netscape "password has expired" "2.16.840.1.113730.3.4.4"
  Netscape "password expiration warning" "2.16.840.1.113730.3.4.5"
  ```

  Note

  Netscape controls are supported by Netscape Directory Server, Netscape/Sun iPlanet, and Red Hat/ Fedora Directory Server.

  Password policy client controls other than the Netscape controls mentioned above are not supported.

  Password expiration warnings will not be seen during OpenVMS login when using directory server software that does not support Netscape password policy client controls, such as Active Directory and Novell eDirectory.

- Characters used in user names and passwords are restricted to the 8-bit ISO 8859-1 (Latin-1) character set. UTF-8 support is not included in this release.

- The **`SET PASSWORD`** command is not supported for SSH logins.

6.2. Mapping Restrictions

- SSH login is not supported for mapped users.

- While executing DECnet operations, such as DECnet copy, you must use the user name and password that is present in the SYSUAF.DAT file.

- The "SYSTEM" account is not mapped for the following scenarios:

  - If a user enters "SYSTEM" at the user name prompt, the user is mapped only to the "SYSTEM" account in SYSUAF.DAT.

  - If the mapping is done for any user to SYSTEM, for example, "johnd" is mapped to "SYSTEM" account in SYSUAF.DAT, then this mapping does not occur and the user gets an Operation failure error at the login prompt.