EVIDEN

Identity and Access Management

DirX Directory

Guide for CSP Administrators

Version 9.1, Edition June 2025

Table of Contents

Preface

The *DirX Directory Directory Guide for CSP Administrators* describes the aspects of installing, configuring, administering and using the Eviden product DirX Directory (DirX) in the context of a Certificate Provisioning Service operating in accordance with regulations like the German “Signaturgesetz”.

- Chapter 1 introduces the material to be covered in the guide.
- Chapter 2 provides information about the installation environment.
- Chapter 3 describes how to configure DirX Directory.
- Chapter 4 provides information about access control.
- Chapter 5 provides information on how to operate DirX Directory.
- Appendix A lists abbreviations and their meanings.

This manual is an enhancement to the DirX Directory document set. Familiarity with this documentation set helps to understand this manual.

DirX Directory Documentation Set

DirX Directory provides a powerful set of documentation that helps you configure your directory server and its applications.

The DirX Directory document set consists of the following manuals:

- *DirX Directory Introduction*. Use this book to obtain a description of the concepts of DirX Directory.
- *DirX Directory Administration Guide*. Use this book to understand the basic DirX Directory administration tasks and how to perform them with the DirX Directory administration tools.
- *DirX Directory Administration Reference*. Use this book to obtain reference information about DirX Directory administration tools and their command syntax, configuration files, environment variables and file locations of the DirX Directory installation.
- *DirX Directory Syntaxes and Attributes*. Use this book to obtain reference information about DirX Directory syntaxes and attributes.
- *DirX Directory LDAP Extended Operations*. Use this book to obtain reference information about DirX Directory LDAP Extended Operations.
- *DirX Directory External Authentication*. Use this book to obtain reference information about external authentication.
- *DirX Directory Supervisor*. Use this book to obtain reference information about the DirX Directory supervisor.
- *DirX Directory Plugins for Nagios*. Use this book to obtain reference information about DirX Directory plugins for Nagios.
- *DirX Directory Disc Dimensioning Guide*. Use this book to understand how to calculate and organize necessary disc space for initial database configuration and enhancing existing configurations.
- *DirX Directory Guide for CSP Administrators*. Use this book to obtain information about installing, configuring and managing DirX Directory in the context of a Certificate Provisioning Service operating in accordance with regulations like the German “Signaturgesetz”.
- *DirX Directory Release Notes*. Use this book to install DirX Directory and to understand the features and limitations of the current release.

Notation Conventions

Boldface type
In command syntax, bold words and characters represent commands or keywords that must be entered exactly as shown.

In examples, bold words and characters represent user input.

Italic type
In command syntax, italic words and characters represent placeholders for information that you must supply.

[ ]
In command syntax, square braces enclose optional items.

{ }
In command syntax, braces enclose a list from which you must choose one item.

In Tcl syntax, you must actually type in the braces, which will appear in boldface type.

|
In command syntax, the vertical bar separates items in a list of choices.

...
In command syntax, ellipses indicate that the previous item can be repeated.

install_path
The exact name of the root of the directory where DirX Identity programs and files are installed. The default installation directory is *userID_home_directory**/DirX Identity* on UNIX systems and **C:\Program Files\DirX\Identity** on Windows systems. During installation the installation directory can be specified. In this manual, the installation-specific portion of pathnames is represented by the notation *install_path*.

1. General Information

This chapter provides general information about DirX Directory for Certificate Service Providers (CSPs) and the scenario in which it operates.

1.1. About DirX Directory for CSP

A CSP that operates according to the German “Signaturgesetz” law can use DirX Directory as a Certificate and Certificate Revocation List (CRL) repository.The DirX Directory server ensures:

- Public read access to the certificates that are publicly available for reading via Lightweight Data Access Protocol (LDAP) (“abrufbar”, according to Signatur Gesetz (SigG).
- Protection against unauthorized reading of certificates that are stored only for evaluation purposes (“nachprüfbar”, according to Signatur Verordnung (SigV).
- Protection against unauthorized modifications of stored certificates.

The DirX Directory product’s X.500 access control features are used to distinguish between certificates and CRLs that are “abrufbar” or “nachprüfbar”.

The special aspects of installing, configuring, administering and using DirX Directory for CSP deal mostly with security and availability of the service.

1.2. About the DirX Directory for CSP Scenario

This section describes an example scenario for using DirX Directory in a CSP environment, including:

- Network hosts and applications
- Users and data

1.2.1. Network Hosts and Applications

The following figure shows the general architecture of a distributed DirX Directory service within a CSP boundary.

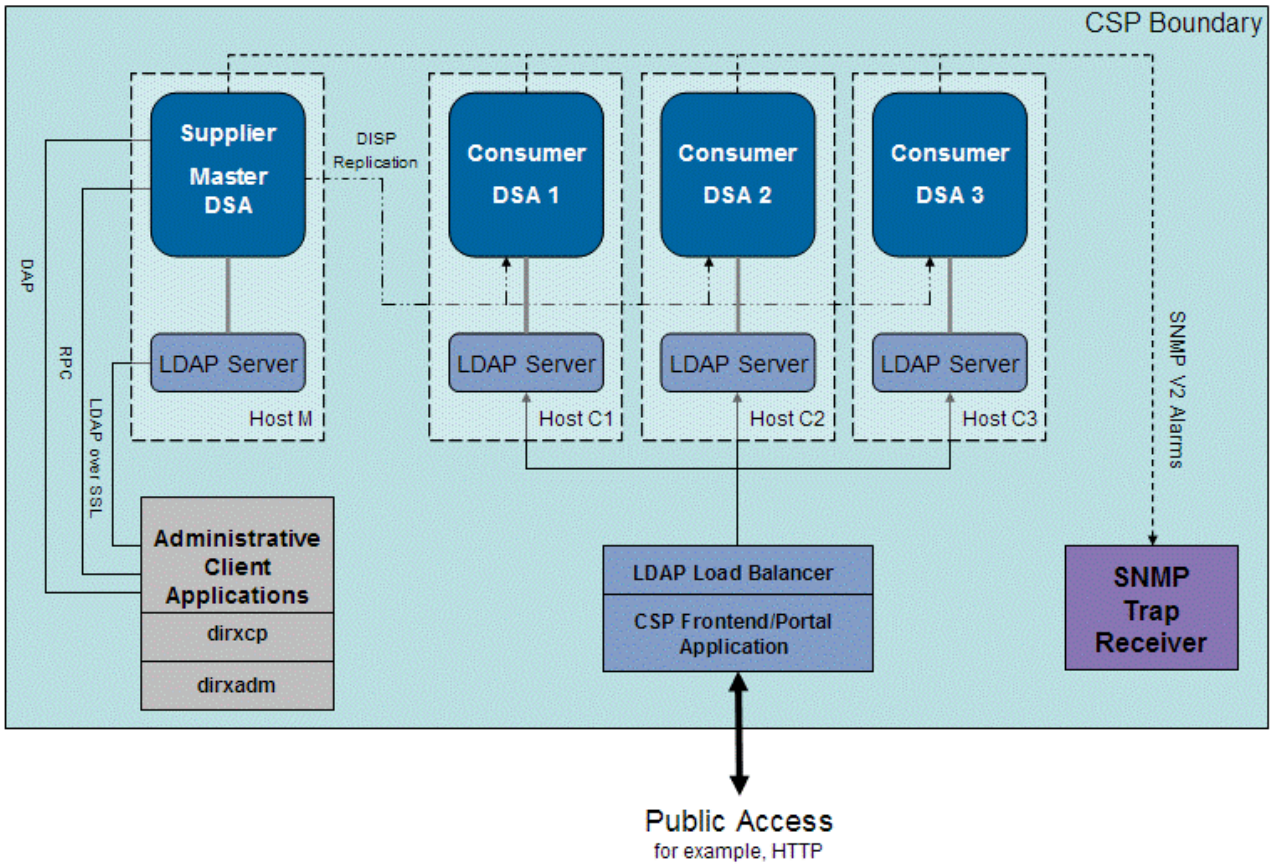

Figure 1. Network Hosts and Applications

As shown in the figure, the distributed DirX Directory service runs on a several hosts within the CSP boundary. Outside access to this system occurs only via a well-defined communication port to a CSP frontend/portal application; for example, via HTTP. Direct access to the DirX Directory service occurs via LDAP. Public users can read certificates or CRLs from the DirX Directory service if and only if they are allowed to read them. With respect to DirX Directory access control, it doesn't matter whether the user accesses the system directly using an LDAP client application or via the CSP frontend/portal application.

The internal network connections of the systems hosting the DirX Directory service should not be visible to the outside world. A firewall product and a virtual private network (VPN) can be used to achieve this state. The provider may also choose to use the SSL/TLS-protected variant of the X.500 protocols (called secure IDM, or IDMS in DirX Directory) for communication inside the CSP boundary.

CSP administrators perform all administrative accesses to the system from inside the CSP boundary. They use DirX Directory DAP or RPC client applications to manage meta-data like the schema, database configuration or access control, and use Ldapv3 over SSL (LDAPS) to manage user data: adding, deleting or changing the status of certificates. We recommend using SSL/TLS client authentication as the only accepted authentication method for these operations.

The hosts on which the DirX Directory servers are running are dedicated to DirX Directory to minimize risk.

For availability and performance reasons, the DirX Directory service consists of master

servers and multiple shadow consumer servers. Read access is distributed onto the consumer servers using an LDAP load balancer.

The DirX Directory service can be configured to send out Simple Network Management Protocol (SNMP) traps (also called alarms) on conditions that require CSP administrator attention, such as security-relevant events and critical operational status.

1.2.2. Users and Data

In the context of this guide, the following directory tree is assumed:

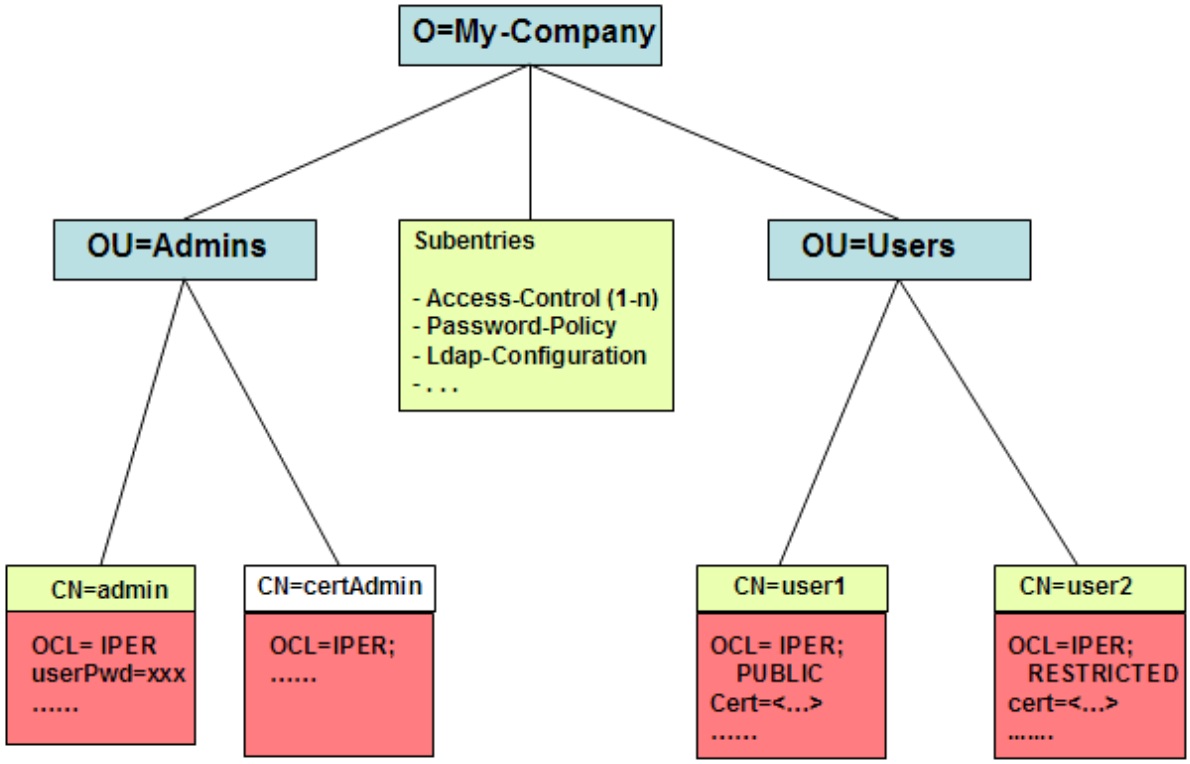

Figure 2. Users and Data

The certificates hosted by the CSP are stored as attributes of user entries in the directory server below the subtree **OU=Users**. On the basis of access control and configuration, the administrator **CN=admin** has the privileges required to administer the entire service, including access control. The CSP frontend/portal application uses the administrator **CN=certAdmin** account to add the certificate attributes and modify the certificate access status (expressed by the objectClass values **OCL=PUBLIC** versus. **OCL=RESTRICTED**). A set of X.500 access control subentries (ACS) is located directly under the root of the tree **O=My-Company**. These subentries ensure that certificates are only readable for anonymous or authenticated users without special privileges if the certificates belong to user entries with the **PUBLIC** objectClass value.

2. Installation and Environment

The DirX Directory installation in the context of the CSP should be limited to the components that are needed for operating the CSP.

The general approach is to install as little software as possible on the machines hosting the DirX Directory service, because each application running on a host increases the risk of inadvertent data exposure. Ideally, the machines hosting the DirX Directory master and consumer servers are dedicated to the DirX Directory service; that is, the only accounts required on the hosts are **root** and **dirx**. Furthermore, “normal” users should not be added to administrator groups.

As shown in the following figure, the DirX Directory server installation DVD provides the following software packages:

- DirX Directory (32-bit version)
- DirX Directory (64-bit version)
- DirX Directory Manager

For performance reasons, we recommend installing the 64-bit version as long as the target hardware can support the 64-bit architecture. Using the 64-bit version (rather than the 32-bit version) allows the DirX Directory server to maintain larger DBAM caches.

We do not recommend installing the DirX Directory Manager package, because working with the command line and scriptable administrative client applications (**dirxadm** and **dirxcp**) contained in the DirX Directory installation package is more reproducible than working with a GUI client.

The installer executables and data files on the product DVD are virus-checked. The associated checksums can be found on the DirX Directory service portal along with patches and service packs. The address of the DirX Directory service portal is:

https://support.dirx.solutions/

3. Configuration

This chapter describes how to configure DirX Directory for CSP, including how to configure:

- SSL/TLS protection for X.500 protocols
- Replication
- Authentication-related attributes and policies
- LDAP servers

3.1. Configuring Encrypted X.500 Protocols

DirX Directory supports performing the X.500 protocol exchanges over a secure IDM stack (IDMS), where the SSL/TLS protocols are applied over the transport layer.This mechanism affects the DAP protocol exchanges initiated by the DUAs (the **dirxcp** client application or the LDAP server process) as well as the DSA-to-DSA protocol exchanges for replication (DISP) and chaining (DSP).

While SSL/TLS allows performing a mutual authentication of the communication peers, the goal of using IDMS in DirX Directory is the encryption of the protocol data exchanged.

Configuring IDMS includes the following tasks:

- Using the addressing in PSAPs to activate IDMS
- Installing individual IDMS key material (optional)

3.1.1. Setting up IDMS Addressing

In DirX Directory, the DNS component of a PSAP address contains the information about whether DAP, DISP or DSP is performed over plain IDM or over protected IDMS.

3.1.1.1. Initializing IDMS

The DNS subcomponent in each communicating process's own address is used to initialize IDMS. This address is defined by:

- The environment variable **DIRX_OWN_PSAP** for the DSA
- The SELF entry in **dirxcl.cfg** and **dirxldap.cfg** for the DUAs

If the address contains an SSLPORT with a value greater than zero, the IDMS stack is initialized. In the DSA process, IDMS starts a communication listener on the port specified in SSLPORT.

The MODE keyword has no meaning in the address and can therefore be omitted.

For example, to initialize the IDMS stack in the LDAP server process on the master directory server host, the LDAP server's **dirxldap.cfg** configuration file must contain a SELF address with an SSLPORT:

Self
TS=Client1,NA='TCP/IP_IDM!internet=1.2.3.4+port=1111',DNS='(HOST=hostM,SSLPORT=21201,PLAINPORT=21200)'

3.1.1.2. Connecting to Communication Peers via IDMS

To connect to a peer DSA, the initiator of the connection uses the DNS subcomponent of the remote DSA's address and its own initialization state.

The remote address is defined by:

- The PSAP from the CK for a shadow supplier DSA as an initiator.
- The PSAP from the SUK for a shadow consumer DSA as an initiator.
- The PSAP from the XR, SUBR or SUPR for a DSA in a distributed directory as an initiator.
- The contact DSA entry in **dirxldap.cfg** for the DUA in the LDAP server process.
- The named DSA entries in **dirxcl.cfg** for the DUA in **dirxcp**.

The relevant parts of the remote DSA's DNS are the MODE and the SSLPORT. If the MODE is set to SSL and the SSLPORT has a value greater than zero, the process connects via IDMS to the specified port of the peer.

For example, to connect securely via IDMS to the DSA, the LDAP server process on the master directory server host must have the following entry in its **dirxldap.cfg** configuration file as the contact DSA:

```
/CN=masterDSA
TS=DSA1,NA='TCP/IP_IDM!internet=1.2.3.4+port=21200',DNS='(HOST=hostM,
SSLPORT=21201,PLAINPORT=21200,MODE=SSL)'
```

For more examples of how the DNS component controls IDMS usage, see the section "Setting up Replication".

3.1.2. Installing IDMS Key Material

IDMS is based on the SSL/TLS security protocols, which require suitable cryptographic key material in order to function.

The DirX Directory product setup installs sample key material that allows for performing IDMS-protected X.500 protocols out of the box. However, users may wish to install their own individual key material.

The **client**, **ldap** and **server** subfolders of the DirX Directory installation folder contain a configuration file named **idmssl.cfg** with the names and locations of the files containing the necessary key material. Here is the content **of idmssl.cfg** as installed:

```
########################################
```

```
# SSL Configuration file for IDM protocol
# each line contains a token and a value
########################################

# the pathname of the file containing the own private key and
# certificate chain in PEM format
idm_ssl_own_pse_file $DIRINST/conf/IdmPSE.pem

# the pathname of the password file for accessing the private key
idm_ssl_pwd_file $DIRINST/conf/IdmPSE.pwd

# the pathname of the PEM file that contains trusted CA certificates
# the file may contain multiple CA certificates in PEM format.
# In client mode, these CA certs are used to verify the certificate
# received from the server.
idm_ssl_trusted_ca_cert_file $DIRINST/conf/testCA.pem

# security protocol to be used - one of: SSLv3, TLSv1, TLSv11, TLSv12
idm_ssl_protocol TLSv12

#  ciphers to use (names must be compatible with OpenSSL naming
schema)
idm_ssl_ciphers HIGH

# max wait time in seconds in SSL I/O
idm_ssl_io_timeout 10

# SSL log level (0=off,1=low)
#  0 = off
#  1 = low    (SSL function calls)
#  2 = medium ( == low + select/poll eventing )
#  3 = high   ( == medium + I/O data )
idm_ssl_logging 0

########################################
```

The variable $DIRXINST shown in the example is replaced with the actual installation path during installation time.

Exchanging the installed example key material with individual keys can be done on a per-host basis; for example, by placing a suitable PEM file with its own private key and the certificate chain in the **conf** subfolder of the installation path for use by all DirX Directory processes (like the sample key material setup).Alternatively, each DirX Directory process

type (**dirxcp**, **dirxldapv3** and **dirxdsa**) can access individual key material files.

3.2. Setting up Replication

Setting up distributed DirX Directory service replication in the CSP environment includes the following tasks:

- Setting up DSA names and PSAP addresses
- Setting up shadowing agreements
- Configuring attribute indexes
- Configuring the CSP frontend/portal application for LDAP server support

The next sections describe these tasks in more detail.

3.2.1. Specifying PSAP Addresses and DSA Names

Each DSA must have a DN and a PSAP address that identifies the DSA remotely and detects its role in a distributed replication scenario. Both of these elements are configured as variables in the file:

install_path/**conf/dirxenv.ini**.

The following excerpt of a configuration file shows the respective lines for the shadow supplier DSA:

```
# DSA name and PSAP address
set DIRX_DSA_NAME=cn=masterDSA
set DIRX_OWN_PSAP=TS=DSA,NA='TCP/IP_IDM!internet=1.2.3.4+port=21200'
,DNS='(HOST=hostM,PLAINPORT=21200,SSLPORT=21201,MODE=SSL)'
```

The following excerpt of a configuration file shows the respective lines for one of the shadow consumer DSAs:

```
# DSA name and PSAP address
set DIRX_DSA_NAME=cn=consumerDSA1
set DIRX_OWN_PSAP=TS=DSA,NA='TCP/IP_IDM!internet=1.2.3.4+port=21200'
,DNS='(HOST=hostC1,PLAINPORT=21200,SSLPORT=21201,MODE=SSL)'
```

The syntactically correct (but unknown) value 1.2.3.4 is specified for the NA component of the Presentation-Address structured attribute because X.500 defines the NA component as mandatory. DirX Directory has added support for the DNS component, which is used with priority. If the DNS component is present in a PSAP, DirX Directory uses the hostname specified and resolves it by means of the DNS to a network address, and

> uses it together with the (plain or SSL) port to connect. Correct DNS configuration is therefore a prerequisite for using the DNS component in PSAP addresses.

3.2.2. Creating the Shadowing Agreements

The floating master setup with multiple hot-standby consumer DSAs ensures high availability of the service for retrieval operations and allows for shutting down servers for maintenance reasons without interrupting the service. In the worst case – a crash of the shadow supplier server – one of the shadow consumer DSAs can take over the supplier role via the emergency switch command.

Using only the DNS component of the PSAP addresses of the participating DSAs creates a central administration point (the DNS server or the **/etc/hosts** file) that defines the physical endpoints of a communication. This setup allows for easily moving a server from one host to another and avoids having to reconfigure clients with the address of the shadow supplier DSA.

Shadowing agreements are administered using the **dirxadm sob** commands. The commands are performed on the shadow supplier DSA. The shadow configuration is stored in a special subentry named **cn=cooperating-dsas-subentry**, which contains entries for all participating DSAs with their DNs, their PSAPs, and their roles in the system.

The following **dirxadm sob create** operation creates the replication setup between the master DSA and consumer DSA 1, as shown in "Figure 1: Network Hosts and Applications".

```
dirxadm> sob create -supplier {/cn=masterDSA} -supplierpsap \
"TS=DSA,NA='TCP/IP_IDM!internet=1.2.3.4+port=21200',DNS=(HOST=hostM,P
LAINPORT=21200,SSLPORT=21201,MODE=SSL)'" \
 -consumer {/cn=consumerDSA1} -consumerpsap \
"TS=DSA,NA='TCP/IP_IDM!internet=1.2.3.4+port=21200',DNS='(HOST=hostC1
,PLAINPORT=21200,SSLPORT=21201,MODE=SSL)'"\
 -consumerkind CENTRALADMIN \
 -agreementid 1 -status cooperative \
 -pol {CONS={REPLS=TRUE}} \
 -agreement "SS={AREA={CP={/O=My-Company},RA={DEF=TRUE}}, \
          ATT={DEF=TRUE}}, \
          UM={SI={OC=TRUE}}" \
```

This operation creates the shadowing agreement with the agreementID **1**, which specifies the shadowing agreement from the master DSA to the consumer DSA 1: The policy option (**-pol**) allows the consumer DSA to become the master via the **sob switch** operation. The agreement is specified as a full shadow, which means that all entries are replicated with all their attributes immediately after the entry is modified on the master DSA to the consumer DSA. The agreement is created with the status **cooperative**, which means that after completing the command, the supplier starts sending the whole directory tree to the

consumer (DISP total update). By specifying **MODE=SSL** in **consumerPSAP**, the agreement is set up to communicate over IDMS.

According to the example in "Figure 1: Network Hosts and Applications", agreements 2 and 3 must be created with analogous **dirxadm** operations.

3.2.3. Configuring the Index

Index setup for shadow suppliers is different from the setup for shadow consumers. On the shadow supplier, an attribute index is not needed or is even inappropriate because the index would need to be updated after the attribute is modified. On the shadow consumer, the attribute index is needed because this is where the search operations are performed. Consequently, the administrator should create the indexes for the **userCertificate** attribute only on the shadow consumer DSAs.

Use the **dirxadm db attrconfig** operation to perform attribute index configuration. The following operation creates suitable indexes for certificate searches. The command must be performed on all three consumer DSAs.

```
dirxadm> db attrconfig uc -index INITIAL ANY
```

3.2.4. Configuring the CSP Application for LDAP Servers

The administrator needs to set up and configure the CSP frontend/portal application so that the LDAP read requests issued on behalf of public users are directed to one of the LDAP servers connected to a consumer DSA.The administrator can use an LDAP load balancer to achieve this configuration; for example, by configuring the list of the IP addresses of Host C1, Host C2 and Host C3 with their respective ports as the addresses to use.These LDAP servers can be configured as read-only servers via their LDAP configuration subentries.

Write access to the directory server should be directed to the LDAP server(s) connected to the shadow supplier DSA.

3.3. Configuring Authentication-related Attributes and Policies

This section describes how to configure authentication-related attributes and policies.

3.3.1. Configuring Administrators

Administrators in DirX Directory are represented by entries that contain a **userPassword** attribute as the credential for the simple LDAP bind operation.Alternatively, the administrators can perform strong authentication (LDAP Simple Authentication and Security Layer (SASL) binds with the mechanism type “EXTERNAL”) based on their Personal Security Environments (PSEs).

The access control items of access control subentries control the administrative privileges of administrators that are needed to perform LDAP operations.There are two other places where privileges must be administered:

a. DirxAdministrators

   The **DirxAdministrators** (DADM) attribute of the root DSE (**/**) controls which authenticated users are allowed to execute operations via the **dirxadm** administrative client operation.

   Adding DNs to the DADM attribute is performed via **dirxadm** as part of the bootstrap process.

   Here is an example of how to add the administrator to the DADM attribute:

   ```
   dirxadm> modify / -add DADM=/O=My-Company/ou=admins/cn=admin
   ```

b. ldapExtOpAdmins

   The **ldapExtOpAdmins** attribute of the LDAP configuration subentry controls which authenticated users are allowed to perform LDAP extended operations. LDAP extended operations allow retrieving configuration and state information about the DirX Directory server processes and performing administrative tasks; for example, enabling and disabling auditing and caching. The **dirxextop** command performs LDAP extended operations.

   Adding DNs to the **ldapExtOpAdmins** attribute is performed via LDAP. DirX Directory also provides more fine-grained access rights to LDAP extended operations. (See section “Attributes Controlling LDAP Extended Operations” of chapter “DirX Directory Attributes” in the *DirX Directory Administrations Reference* for details.)

   Here is an example of how to add an administrator to the **ldapExtOpAdmins** attribute with the **dirxcp** command:

   ```
   dirxcp> modify CN=ldapConfiguration,O=My-Company \
     -add ldapextopadmins=CN=admin,OU=Admins,O=My-Company
   ```

3.3.2. Configuring DSA Policy

The DirX Directory DSA policy describes - among other options - the authentication methods and the DSA passwords used for the DSA-to-DSA bind operations occurring in the replication protocol (DISP) and in the chaining protocol (DSP). The DSA policy is stored in the DSA-Policy attribute (**DSAP**) of the X.500 root entry (**/**) and is administered with the **dirxadm** command.

In the context of CSP operation, we recommend that all DSAs perform binds with individual DSA passwords using the Simple-Protected X.500-defined mechanism.

The following set of **dirxadm** operations instantiates the DSA policies. It must be applied to all participating DSAs (master and all consumer DSAs):

```
dirxadm> modify / -add DSAP={DSA={/CN=MasterDSA},            \
                             AP={AMB={AM=SIMPLE-PROTECTED},  \
                                 PWD=masterPWD             \
                                }                           \
                            };

dirxadm> modify / -add DSAP={DSA={/CN=ConsumerDSA1},         \
                             AP={AMB={AM=SIMPLE-PROTECTED},  \
                                 PWD=consumer1PWD            \
                                }
                            };

dirxadm> modify / -add DSAP={DSA={/CN=ConsumerDSA2},         \
                             AP={AMB={AM=SIMPLE-PROTECTED},  \
                                 PWD=consumer2PWD            \
                                }                           \
                            };
```

The DSA policy should be configured before the shadow agreements are established as part of the replication setup.

3.3.3. Configuring User Policy

The DirX Directory user policy describes the authentication methods allowed for particular users, expressed by subtrees or specific DNs. The user policy is stored in the User-Policy attribute (**USP**) of the X.500 root entry (**/**) and is administered with the **dirxadm** command.

In the context of CSP operation, the user policy forces particular users to perform strong authentication methods when binding to the DirX Directory service.

The following **dirxadm modify** operation provides an example of how the certificate administrator can be forced to authenticate using a strong bind method.

```
dirxadm> modify / -add USP={USN={/O=My-Company/CN=certAdmin}, \
AP={AMB={AM=STRONG}}}
```

The effect of this command is that the user named **CN=certAdmin,O=My-Company** is unable to bind to the directory unless he uses the SSL/TLS client authentication, which is the only bind mechanism supported by DirX Directory with a STRONG authentication level. This method requires that the user owns a Personal Security Environment (PSE) and that

the issuer of the PSE is trusted by the DirX Directory server. (See the description of the LDAP server SSL configuration attribute **ldapTrustedCACerts** in section 3.4.3, "Configuring the Additional LDAP Server on the Master" for details.) The following **dirxcp bind** operation provides an example for an SSL/TLS client authentication:

```
dirxcp> bind -prot ldapv3 -sasl -mech EXTERNAL -certsub certAdmin
-key3pass PSE-passwphrase -address localhost:1636
```

When this user policy is applied, any other type of bind will not succeed for the user with the DN **CN=certAdmin,O=My-Company**, while other users are unaffected.

3.3.4. Configuring Password Policy

DirX Directory password policy supports the users and administrators in choosing secure bind passwords. The password aging feature ensures that passwords are changed within the specified time frame. The account-locking feature is intended to prevent dictionary attacks on passwords. Password policies are stored as special X.500 subentries in DirX Directory. They can be created using the **dirxcp obj create** operation.

The following **dirxcp create** operation creates an example password policy subentry:

```
dirxcp> create CN=CSP-PwdPol,O=My-Company -attr \
{subtreeSpecification;binary=MIAAAA==} \
{ocl=pwdPolicy;subEntry} pwdMinAge=3600 pwdMaxAge=15552000
pwdInHistory=10 pwdCheckQuality=2 pwdMinLength=8 \
pwdMinSpecialChar=2 pwdMaxLength=10 pwdExpireWarning=864000 \
pwdLockout=TRUE pwdMaxFailure=3 \
pwdFailureCountInterval=120 pwdMustChange=TRUE \
pwdExclusions=CN=admin,O=My-Company \
```

The effect of this subentry is reported by the DirX Directory DSA during startup of the service by the following message:

```
DSA Password Policy Info:
Password Policy Settings:
Relevant Subentry: cn=CSP-PwdPol,o=My-Company
 Password Syntax check:    Enabled, reject hashed Passwords
  Minimal Password Length:  8
  Minimal Number of specials: 2
  Maximal Password Length:  10
  Password History Size:   10
 Password Aging:
```

```
 Password expires after:   15552000 sec (180 days)
 Warning sent before Expiry: 864000 sec (10 days)
 Grace Logins after Expiry: None
 Minimal Password Age:    3600 sec (1 hours)
Account Lockout:       Enabled
 Lockout Duration:      Forever (until reset by admin)
 Account locked after:    3 consecutive failed logins
              within 120 sec
Password Change after Reset: Required
Password Policy Exclusions:  1 Entry/Node
 DN: CN=admin,O=My-Company
```

That is, all users that perform password-based bind operations with the exception of the administrator named **CN=admin,O=My-Company** must change their passwords every 180 days, the password must be conform to the syntax rules and the 10 most recent passwords must not be used. Three attempts to bind with incorrect passwords within 120 seconds cause the account to be locked. The administrator must unlock the account explicitly.

3.3.5. Disabling External Authentication

DirX Directory can be configured to forward incoming simple authenticated bind operations over LDAP to an external authentication service such as the Windows Domain Controller or to an external LDAP server.Bind forwarding should not be used in the context of a CSP.

To disable external authentication, the configuration file

*install_path***/server/conf/dirxextauth.cfg**

must not exist.This is the default condition after installing the product.

3.4. Configuring the LDAP Servers

A set of subentries specifies the configuration of the DirX Directory LDAP server.Unless default values should be used, each LDAP server process needs the following types of subentries:

- A subentry with the objectClass **ldapConfiguration**.This subentry stores the LDAP server's general configuration and contains attributes that control, for example, the ports on which the LDAP server listens, the serviceControls applied to the DAP requests sent to the DSA, properties like read-only, LDAP client IP addresses and users denied or allowed to bind to the server and many more.
- A subentry with the objectClass **ldapSSLConfiguration**.This subentry stores the LDAP server's SSL/TLS security configuration; for example, the LDAP server's own key material and whether or not the LDAP server requires SSL/TLS client authentication.
- A subentry with the objectClass **ldapAuditConfiguration**.This subentry stores the LDAP

server's auditing configuration; for example, the strategy to be applied after an audit file exceeds its maximum size or the degree of detail of the audit records.

Because the LDAP server configuration subentries are subject to replication (as is every entry or subentry within a replication area), the administrator only needs to administer the configuration subentries on the master server.The replication process then distributes all subentries to all consumer servers.

In the context of the scenario shown in Figure 1: Network Hosts and Applications, a single ldapAuditConfiguration subentry controls the audit configuration of all the LDAP servers because there is no need to have a different audit behavior.See section 5.1.2 "Enabling LDAP Audit" for details on creating the LDAP audit subentry.

For the general configuration and the SSL aspects of the LDAP servers, there are different requirements on the particular hosts. Therefore there are individual sets for ldapConfigurations and ldapSSLConfigurations. The LDAP server processes on the hosts will find their individual configuration by matching the commonName of the ldapConfiguration subentry with the value of the environment variables

```
DIRX_DEFAULT_LDAP_SERVER or DIRX_ADDITIONAL_LDAP_SERVERS
```

3.4.1. Configuring the Shadow Servers

The shadow server hosts are "HostC1" and "HostsC3". Configuring these servers means specifying the attributes and values of the configuration subentries **cn=lcfgShadow** and **cn=lcfgSSLShadow**. The configuration specifies read-only LDAP servers listening on ports 389 (LDAP) and 636 (LDAPS). LDAP extended operations are prohibited. The SSL/TLS-secured access does not require client authentication via the LDAP SASL bind.

By setting the following variable in the configuration file *install_path***/conf/dirxenv.ini**:

```
set DIRX_DEFAULT_LDAP_SERVER=lcfgShadow
```

the LDAP server process on these hosts use the individual configuration subentries.

In detail, the configuration subentries are created with the following **dirxcp** (LDAP) operations:

```
dirxcp> create CN=lcfgShadow,O=My-Company -attr \
{subtreeSpecification;binary=MIAAAA==} \
{ocl=ldapConfiguration;subentry} \
ldapPortNumber=389 \
ldapSecurePortNumber=636 \
ldapReadOnlyServer=TRUE \
ldapExtOpAdmins=none \
```

```
ldapSSLCfgSubentryCN=ldapSSLConfiguration \
ldapAuditCfgSubentryCN=CommonldapAudit

dirxcp> create CN=ldapSSLConfiguration,O=My-Company -attr \
{subtreeSpecification;binary=MIAAAA==} \
{ocl=ldapSSLConfiguration;subentry} \
supportedEncryptionStrength=HIGH
{supportedSecurityProtocols=SSLV3.0;TLS1.0}
requireSSLClientAuth=FALSE \
ldapOwnKeyMaterialPEM_FILE=url of the PEM file with key material
```

The **supportedSecurityProtocol** setting depends on the capabilities of the LDAP client—the CSP Front End/Portal application—that connects to the LDAP server; for example, JRE 7-based LDAP client applications support newer versions of TLS, like TLSv1.1.

3.4.2. Configuring the Default LDAP Server on the Master

The default LDAP server runs on the host “HostM”. Configuring the default LDAP server means specifying the attributes and values of the default configuration subentries **cn=ldapConfiguration** and **cn=ldapSSLConfiguration**. The configuration specifies a read-write LDAP server listening on ports 389 (LDAP) and 636 (LDAPS). The entry named **CN=admin,OU=Admins,O=My-Company** is allowed to perform LDAP extended operations. The SSL/TLS secured access does not require client authentication via the LDAP SASL bind.

Because the default subentries are used, no environment variable must be set to publish the common name of the configuration subentry.

In detail, the configuration subentry is created with the following **dirxcp** (LDAP) operation:

```
dirxcp> create CN=ldapConfiguration,O=My-Company -attr \
{subtreeSpecification;binary=MIAAAA==} \
{ocl=ldapConfiguration;subentry} \
ldapPortNumber=389 \
ldapSecurePortNumber=636 \
ldapReadOnlyServer=FALSE \
ldapExtOpAdmins=CN=admin,OU=Admins,O=My-Company \
ldapSSLCfgSubentryCN=ldapSSLConfiguration \
ldapAuditCfgSubentryCN=CommonldapAudit
```

3.4.3. Configuring the Additional LDAP Server on the Master

An additional LDAP server process is to be started on host “HostM” that requires SSL client

authentication. This configuration is achieved by exporting the following environment variable that publishes the name of the configuration subentry for the additional LDAP server:

```
set DIRX_ADDITIONAL_LDAP_SERVERS=lcfgSasl+7999
```

The corresponding subentries are named **cn=lcfgSasl** and **cn=lcfgSSLSasl**. The configuration specifies a read-write LDAP server listening only on the LDAPS port 1636. LDAP extended operations are prohibited. The SSL/TLS secured access requires client authentication via the LDAP SASL bind.

This LDAP server is intended to perform the operations initiated by the certificate administrator named **CN=certAdmin,OU=Admins,O=My-Company**.

In detail, the configuration subentries are created with the following **dirxcp** (LDAP) operations:

```
dirxcp> create CN=lcfgSasl,O=My-Company -attr \
{subtreeSpecification;binary=MIAAAA==} \
{ocl=ldapConfiguration;subentry} \
ldapPortNumber=0 \
ldapSecurePortNumber=1636 \
ldapReadOnlyServer=FALSE \
ldapExtOpAdmins=none \
ldapSSLCfgSubentryCN=lCfgSSLSasl \
ldapAuditCfgSubentryCN=CommonldapAudit

dirxcp> create CN= lCfgSSLSasl,O=My-Company -attr \
{subtreeSpecification;binary=MIAAAA==} \
{ocl=ldapSSLConfiguration;subentry} \
supportedEncryptionStrength=HIGH
{supportedSecurityProtocols=SSLV3.0;TLS1.0}
requireSSLClientAuth=TRUE \
ldapOwnKeyMaterialPEM_FILE=url to the PEM file with key material \
ldapTrustedCACerts_FILE=url to CA certificate file trusted to issue
user certs
```

4. Access Control

The DirX Directory access control mechanism is the basis for distinguishing whether certificates are stored publicly readable (“abrufbar”) or not (“nur nachprüfbar”).

Certificates are stored as attributes of directory entries. For each entry, the X.500 access control allows for defining WHO is allowed to perform WHAT kind of operation on WHICH data, where:

- WHO is the authenticated entity that performed the bind operation; for example, the user with a specific DN or the anonymous user.
- WHAT is the LDAP operation performed on the data; for example, a read or a modify operation.
- WHICH data is the target of the operation, which is always an entry with an attribute set; for example, read entry with a specific DN with the attributes “Certificate” and “email”.

DirX Directory supports multiple ways to specify the access control:

- Access control subentries holding Prescriptive-ACI attributes with access rules that apply for all entries in a particular subtree or a specific area of a particular subtree.
- Access control subentries holding Prescriptive-ACI attributes with access rules that apply for all entries within a subtree that fulfill a particular ObjectClass filter condition (Specification Filter SF).
- Entry-ACI attributes stored at each individual entry with access rules that apply for this particular entry only (EACI).

The following sections describe how to apply the second method (SF), because this option has the following advantages for a CSP:

- An LDAP application - for example, the CSP frontend/portal application - can easily read and parse the objectClass value. These values are handled in LDAP as simple strings. The application can use an administrative account to bind to the directory and then operate on behalf of some “normal” user. The objectClass value specifies whether or not the normal user is allowed to access the certificate.
- A simple modify operation (over LDAP) replacing an objectClass value is used to change the state of a certificate (for example, from “abrufbar” to “nur nachprüfbar“).It does not require moving the entry into another subtree or modifying the (rather complex) EACI.

4.1. Defining the objectClasses

The objectClass value serves two purposes:

- It triggers the access control and thereby protects the certificate from unauthorized access.
- It contains the status with respect to access control in an LDAP-readable format.

The best way to modify the schema is to load an LDIF change file with the **dirxload** command.The following excerpt of an LDIF change file creates the schema extension with the two auxiliary objectClass values **OcPublic** and **OcRestricted** for use within the access control on certificate attributes.They specify whether "normal" users can read ("abrufbar") (objectClass **OcPublic**) or just compare ("nur nachprüfbar") (objectClass **OcRestricted**) certificates:

```
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: (1.3.12.2.1107.1.3.6.201 NAME 'OcPublic' DESC 'This
object class marks certificates as abrufbar' SUP top AUXILIARY)
objectClasses: (1.3.12.2.1107.1.3.6.202 NAME 'OcRestricted' DESC
'This object class marks certificates as nur nachpruefbar' SUP top
AUXILIARY)
```

4.2. Creating Access Control Subentries

To separate certificates into publicly-readable and comparison-only categories, access control subentries must be added to the root entry of the subtree to apply these rules.Then the auxiliary objectClass **OcPublic** or **OcRestricted** of each entry in this subtree determines access to the certificate attributes.Access control subentries are added with the **dirxcp** command over DAP.In the following example, two access control subentries are added to the subtree root entry **o=my-company**:

- **cn=AC-restricted**
  This subentry specifies that
- Only the certificate administrator **/O=My-Company/OU=admins/CN=cert-adm** can modify the certificate value and status.
- “Normal” users can only compare (“nur nachprüfbar”) certificates of entries with the auxiliary objectClass **OcRestricted**.
- **cn=AC-public**
  This subentry specifies that
- Only the certificate administrator **/O=My-Company/OU=admins/CN=cert-adm** can modify the certificate value and status.
- “Normal” users can read (“abrufbar”) certificates of entries with the auxiliary objectClass **OcPublic**.

```
create "O=My-Company/CN=AC-restricted" -attr \
    {OCL=SUBE;ACS} \
    {SS={BAS={/},SF={ITEM=OcRestricted}}} \
    {PACI={ID=certAdmin - enable Cert Modification and Read,
```

```
       PR=255,
       AL={BL={L=STRONG}},
       UF={UC={N={DN={/O=My-Company/OU=Admins/CN=certAdmin}}},
         UP={PI={E=TRUE},
GAD=grantDiscloseOnError+grantRead+grantBrowse+grantReturnDN+grantExp
ort+grantImport+grantModify};
           {PI={AUATV=TRUE},

GAD=grantRead+grantCompare+grantFilterMatch+grantAdd+grantModify+gran
tRemove};
           }
       };
       {ID=Public Access: enable Cert compare - disable Cert Read,
       PR=0,
       AL={BL={L=NONE}},
       UF={UC={AU=TRUE},
         UP={PI={E=TRUE},
           GAD=grantDiscloseOnError+grantRead+grantBrowse
+grantReturnDN};
           {PI={AUATV=TRUE},
           GAD=grantRead+grantCompare+grantFilterMatch};
           {PR=53,
           PI={AT=UC;CRL;ARL;DRL},
           GAD=denyRead+grantCompare+grantFilterMatch}
         }
       }
    }

create "O=My-Company/CN=AC-public" -attr \
    {OCL=SUBE;ACS} \
    {SS={BAS={/},SF={ITEM=OcPublic}}} \
    {PACI={ID=certAdmin - enable Cert Modification and Read,
       PR=255,
       AL={BL={L=STRONG}},
       UF={UC={N={DN={/O=My-Company/OU=Admins/CN=certAdmin}}},
         UP={PI={E=TRUE},
GAD=grantDiscloseOnError+grantRead+grantBrowse+grantReturnDN+grantExp
ort+grantImport+grantModify};
           {PI={AUATV=TRUE},

GAD=grantRead+grantCompare+grantFilterMatch+grantAdd+grantModify+gran
```

```
tRemove};
            }
        };
        {ID=Public Access: enable Cert Read,
        PR=0,
        AL={BL={L=NONE}},
        UF={UC={AU=TRUE},
          UP={PI={E=TRUE},
            GAD=grantDiscloseOnError+grantRead+grantBrowse
+grantReturnDN};
            {PI={AUATV=TRUE},
            GAD=grantRead+grantCompare+grantFilterMatch};
            {PR=53,
            PI={AT=UC;CRL;ARL;DRL},
            GAD=grantRead+grantCompare+grantFilterMatch}
          }
        }
    }
```

Both access control subentries have two PACI attributes:

- One that applies to the certificate administrator **CN=certAdmin,OU=Admins,O=My-Company**.When performing a bind operation using STRONG authentication, this user can read and modify certificates and modify the objectClass values.
- One that applies to the rest of the world.These “normal” users can read certificates of entries with the auxiliary objectClass **OcPublic** and compare certificates of entries with the auxiliary objectClass **OcRestricted**.

According to the X.500 specifications, access is denied by default if there is no explicit grant.As a result, there must be a general access control subentry that rules the access control on the entire subtree without depending on objectClasses.This subentry typically includes access rights for administrators.

4.3. Changing Certificate Access Status

The certificate of an entry is made publicly readable by adding the auxiliary objectclass value **OcPublic**.The following **dirxcp** command is an example of how to change the access state using LDAP:

```
dirxcp> modify CN=user1,OU=Users,O=My-Company \
```

```
                    -add objectClass=OcPublic
```

After performing this command, the userCertificate attribute of entry **cn=user1** is publicly readable.

The command to change the state of an entries certificate from “abrufbar” to “nur nachprüfbar” looks like the following example:

```
dirxcp> modify CN=user1,OU=Users,O=My-Company      \
                    -rem objectClass=OcPublic      \
                    -add objectClass=OcRestricted
```

After performing this command, the userCertificate attribute of entry **CN=user1** can no longer be read by users other than **CN=certAdmin**.

The commands only execute successfully after the user **CN=certAdmin,OU=Admins,O=My-Company** has performed the bind operation using STRONG authentication; that is, a bind operation over LDAP with the options
-sasl and **-mechanism EXTERNAL**.

5. Operation

This chapter describes how to operate DirX Directory in the CSP environment, including how to:

- Administer DSA and LDAP server auditing
- Use SNMP traps
- Check database consistency
- Back up the database
- Manage DirX Directory patches

5.1. Managing Auditing

The DirX Directory DSA and LDAP server processes can be configured to write audit trails containing all key arguments of each protocol operation.The audit files are written in binary format and should be evaluated before archiving them.

In the context of a CSP, the analysis of audits can help to detect issues like performance problems or security breaches.As a result, we recommend enabling the audit feature for both processes with the maximum degree of detail.

This section describes how to enable DSA and LDAP server audit and how to process them for archiving.

5.1.1. Enabling DSA Audit

The following **dirxadm** command enables the DSA's audit facility.After writing 50,000 records in the current audit trail file, the file is moved by appending a timestamp to its filename.The audit detail is set to the maximum the level **attrval** (attributes and values are recorded).The settings are made persistent in the DirX Directory database so that after a process restart, audit writing is continued:

```
dirxadm> audit modify -status on -level attrval -overflow move
```

5.1.2. Enabling LDAP Audit

The following **dirxcp** command creates a new LDAP audit subentry that stores the audit settings for the LDAP server persistently in the DirX Directory database. After a process restart, audit writing is continued:

```
dirxcp> create CN=CommonldapAudit,O=My-Company \
    -attr {objectClass=ldapAudit;subentry} \
    {subtreeSpecification;binary=MIAAAA==} \
    ldapAuditOn=TRUE ldapAuditSizeLimit=100000 \
```

```
    ldapAuditOverFlow=moveFile ldapAuditLevel=max \
    ldapAuditValueLimit=256 ldapAuditOpSelection=all \
    ldapAuditEncryption=none
```

This LDAP audit subentry is relevant for the entire subtree due to its subtreeSpecification attribute.

The subentry configures the LDAP audit facility to move the file after writing 100,000 audit records with the maximum degree of detail.All LDAP operations are subject to auditing.The attribute values from the LDAP requests are written up to the first 256 characters, which is sufficient because the certificates and CRLs are handled as binary attributes in LDAP and thus cannot be written in clear text in the audit records.

5.1.3. Archiving Audit Files

The DirX Directory server processes write audit records in a binary format into the audit files.These files need to be processed off-line using **dirxauddecode** command.Note that the binary format of the audit files may change with a new DirX Directory version.It is therefore necessary to archive the processed audit file rather than the binary-formatted files.

The **dirxauddecode** command supports a number of options to filter the output.

As a preferable choice for the LDAP audits, **dirxauddecode** can be used to produce a comma-separated list of values for each audit record, which makes the audit records easy to parse by programs.Use the LDAP audit configuration file to configure the type of the values reported for a single operation.The **dirxauddecode** command uses this configuration file to customize the output file when evaluating LDAP audit log files.

5.2. Using SNMP Traps

All DirX Directory server processes and the watchdog process can be configured to send out SNMPv2 traps.According to the SNMP specification, the traps are transmitted using UDP/IP and can be trapped by a trap receiver application which is by default listening on port 162. An environment variable controls whether or not SNMP traps are enabled.The different types of traps can be configured individually in a configuration file.

DirX Directory can generate about 40 different traps that can be categorized as follows:

1. Traps that indicate normal state change of the service; for example, starting a process, completing the DBAM cache preload phase and so on.
2. Traps that indicate security-relevant issues; for example, password policy-driven account locking due to multiple incorrect password inputs.
3. Traps that indicate a critical operation status; for example, resource shortage, operation response times exceeding a configurable threshold or unexpected restarts of processes.

In the context of a CSP operation, we recommend activating at least the SNMP trap facility for categories 2 and 3. The *install_path***/conf/snmptraps.cfg** configuration file is a template for SNMP settings, with all traps commented out. The following sections provide a short

discussion of the traps and the trap parameters that a CSP can use.

The DirX Directory SNMP trap facility must generally be enabled by exporting the environment variable

```
DIRX_SNMP=1
```

in the configuration file:

install_path/**conf/dirxenv.ini**.

5.2.1. Security-related Traps

```
trap   dirxUnauthorizedAdminAccess     0      0
```

This trap is sent each time a user tries to bind with the administrative client application **dirxadm** without being configured as the administrator in the dirxAdminstrators (DADM) attribute of the root DSE.

```
trap   dirxAccountLocked               0      0
```

This trap is sent each time a user account is locked due to the password policy settings. In the example above, this is the case if a user sends an incorrect password in the bind operation three times within two minutes. (See section 3.3.4 " Configuring Password Policy" for information about password policy settings.) The administrator must unlock such a user account explicitly.

```
trap   dirxDsaBindProblem              0      0
```

This trap is sent each time a DSA-to-DSA bind operation (DSP or DISP) fails. This may be the case because of a configuration error (individual passwords of DSAs do not match in the respective DSA policy attributes of the root DSEs) or because the host clocks are not synchronous.

```
trap   dirxPduDecodeError              0      0
```

This trap is sent each time the DirX Directory servers cannot decode an operation request. This may be the case because a client application intentionally sends invalid PDUs. In the context of CSP operation, it may result from the attempt to load invalid encoded userCertificates.

5.2.2. Critical State-related Traps

```
trap  dirxServiceStartFailure#     0   0
```

This trap is sent if the DirX Directory watchdog process is unable to start the service processes. This is a fatal situation. Further investigation in the DirX Directory exception files and the system log files is required.

```
trap  dirxProcessCrash        0   0
```

This trap is sent if the DirX Directory watchdog detects the crash of one of its supervised server processes. Although the crashed process is restarted automatically, multiple occurrences of such traps need further investigation in the DirX Directory exception files.

```
dirxThreadLimitReached         5   60
```

This trap is sent—at most five times within one minute—if one of the DirX Directory processes exceeds its configured thread limit. The DSA rejects all subsequent protocol operations. Multiple occurrences of such traps need further investigation in the DirX Directory exception files.

```
trap   dirxThreadPoolExhausted         1        1
```

This trap is sent—at most once per second—if the DirX Directory LDAP server process has no more free operation threads that are able to perform an operation. In this case, the operation is queued in the operation stack. Although the ThreadPoolSize can be increased in the LDAP configuration subentry, multiple occurrences of such traps need further investigation in the DirX Directory exception files.

```
trap   dirxOpStackLimitReached         1      1
```

This trap is sent—at most once per second—if the operation stack of the DirX Directory LDAP server process has no more free slots to queue an incoming LDAP operation. In this case, the LDAP operation is rejected with the error code “server reject”. Multiple occurrences of such traps need further investigation in the DirX Directory exception files.

```
trap   dirxCtxLimitExceeded            5      60
trap   dirxCtxULimitExceeded           5      60
```

These traps are sent—at most five times per minute—if the configured maximum main memory assigned to one operation (CtxLimit) or to all operations (CtxULimit) is exhausted.

Possible reasons include search results that are too large or too many in-parallel paged search operations that are not completed by the client application in time. Although the respective limits can be increased by increasing the values of specific environment variables, multiple occurrences of such traps need further investigation in the DirX Directory exception files.

```
trap   dirxDispAgreementDisabled       0      0
trap   dirxDispAgreementTerminated     0      0
```

These traps are sent every time the DirX Directory DSA disables or terminates a shadowing agreement due to some internal problem. The problem is described in more detail in the DSA exception file. A disabled shadowing agreement may be enabled manually after resolving the problem. A terminated shadowing agreement must be re-established.

```
trap        dirxOpTimeWatch                 0      0
trapparam   dirxLdapOpTimeWatchDuration            20
trapparam   dirxLdapOpTimeWatchFreq                3
```

This trap provides a monitoring of the LDAP operation runtime. In this example, the trap is initiated if an active LDAP operation has been running for longer than 20 seconds. The trap handler checks for such operations every three seconds. Such long-running operations may be observed due to missing attribute indexes. Check the LDAP audit files for the attribute index settings and the long-running operations.

```
trap   dirxDBAMDeviceNearLimit         1      60
```

This trap is sent—at most once per minute—if any of the DBAM devices exceeds a ratio of more than 90%.If the database has not already exceeded its final size, a re-sizing of the DBAM devices using the **dbamconfig** command is required.This step involves a save and restore of the database.

```
trap   dirxDBAMAvidxNearClusterLimit   1   60
```

This trap is sent—at most once per minute—if the attribute indexes require more than 90% of the available index cluster space in the DBAM device.Either an attribute index reorganization (using the **dirxadm db attrconfig -build** operation) or resizing of the DBAM devices (using the **dbamconfig** command) is required.

5.3. Checking Database Consistency

You can use the **dbamverify** command-line tool on a regular basis to check the consistency of the DirX Directory DBAM database.The **dbamverify** command can check:

- The referential integrity of all attribute value blocks and tree blocks
- The consistency of all directory-specific entries (DSEs)—each DSE is read in the context of this check and is verified with respect to DBAM level consistency, linkage of the entry in the tree, and ASN.1 encoding, schema and index consistency of its attributes
- Attribute value index bit strings
- Subordinate index bit strings

Specifying no option directs **dbamverify** to perform all consistency tests.The command returns an exit code of **0** on success or a **1** if it encountered an error.The text of the error message is displayed on **stderr**.

Because **dbamverify** locks the database for update operations while it runs, we recommend that you take a backup of the database with **dirxbackup** and then run **dbamverify** on this archive off-line.The section “Managing Backups” provides an example of this approach.

For details on **dbamverify** and **dirxbackup** command-line syntax, see the *Administration Reference*.

5.4. Managing Backups

Saving backups of the directory’s database is one of the most important tasks of CSP operation.

You must use the **dirxbackup** command to save and restore a database or verify the consistency of a database archive that you created with **dirxbackup**.Saving a directory database does **NOT** require setting the directory in read-only mode, so it does not interfere with normal CSP operation.

The **dirxbackup** command supports both full and delta backups.To restore a saved database, first restore the full backup archive and then restore all delta backups in the correct sequence.For details on **dirxbackup** command-line syntax, see the *Administration Reference*.

Do not use a non-DirX backup tool to save DBAM database files or devices.It can produce inconsistencies in the database and / or in the dirxbackup archive file.

Here is a sample backup and database consistency-checking scenario that can be used in the context of CSP operations:

- Create a full backup every 24 hours—for example, at 12:00 pm—by creating a scheduled job that performs the following **dirxbackup** command:

```
dirxbackup –S dirxDB.date_Time
```

- Create delta backups every *n* hours; for example, every four hours. The actual frequency of delta backups depends on your requirements. To achieve this, create a scheduled job that performs the following **dirxbackup** command:

```
dirxbackup -Sd dirxDB.date_Time.delta.number
```

- Check the archives for internal integrity and completeness by performing:

```
dirxbackup -T dirxDB.date_Time dirxDB.date_Time.number
```

- If the checks on the archives perform successfully, perform the DBAM database consistency check on the archives with the **dbamverify** command. Specify the full backup archive first, followed by the delta backup archives in sequence, for example:

```
dbamverify dirxDB.date_Time dirxDB.date_Time.1 \
    dirxDB.date_Time.2 …
```

5.4.1. Managing Patches

The operators responsible for the CSP service must ensure that the productive directory service always uses the latest released patch of the DirX Directory software.

Appendix A: Abbreviations

This appendix lists abbreviations.

ACI	Access Control Item
CA	Certification Authority
CK	Consumer Knowledge
CN	CommonName
CRL	Certificate Revocation List
CSP	Certification Service Provider
DAP	Directory Access Control
DBAM	Database Access Method
DISP	Directory Information Shadowing Protocol
DN	Distinguished Name
DSA	Director Service Agent
DSAP	DSA Policy (Attribute Abbreviation)
DSE	Directory Specific Entry
DSP	Directory System Protocol
EACI	Entry Access Control Item (Attribute Abbreviation)
IDM	Internet Direct Mapping protocol
IDMS	IDM protected by SSL/TLS
OCL	ObjectClass (Attribute Abbreviation)
LDAPS	Ldapv3 over SSL
OCSP	Online Certificate Status Protocol
PACI	Prescriptive ACI (Attribute Abbreviation)
PDU	Protocol Data Unit
PSAP	Presentation Address
PSE	Personal Security Environment
RDN	Relative Distinguished Name
RPC	Remote Procedure Call
SACI	Subentry ACI (Attribute Abbreviation)
SigG	Signatur Gesetz
SigV	Signatur Verordnung
SNMP	Simple Network Management Protocol
SSL	Secure Sockets Layer
SUB	Subordinate Reference

SUK	Supplier Knowledge
SUP	Superior Reference
TLS	Transport layer Security
VPN	Virtual Private Network
XR	Cross Reference

DirX Product Suite

The DirX product suite provides the basis for fully integrated identity and access management; it includes the following products, which can be ordered separately.

DirX Identity

DirX Identity provides a comprehensive, process-driven, customizable, cloud-enabled, scalable, and highly available identity management solution for businesses and organizations. It provides overarching, risk-based identity and access governance functionality seamlessly integrated with automated provisioning. Functionality includes lifecycle management for users and roles, cross-platform and rule-based real-time provisioning, web-based self-service functions for users, delegated administration, request workflows, access certification, password management, metadirectory as well as auditing and reporting functionality.

DirX Directory

DirX Directory provides a standards-compliant, high-performance, highly available, highly reliable, highly scalable, and secure LDAP and X.500 Directory Server and LDAP Proxy with very high linear scalability. DirX Directory can serve as an identity store for employees, customers, partners, subscribers, and other IoT entities. It can also serve as a provisioning, access management and metadirectory repository, to provide a single point of access to the information within disparate and heterogeneous directories available in an enterprise network or cloud environment for user management and provisioning.

DirX Access

DirX Access is a comprehensive, cloud-ready, scalable, and highly available access management solution providing policy- and risk-based authentication, authorization based on XACML and federation for Web applications and services. DirX Access delivers single sign-on, versatile authentication including FIDO, identity federation based on SAML, OAuth and OpenID Connect, just-in-time provisioning, entitlement management and policy enforcement for applications and services in the cloud or on-premises.

DirX Audit

DirX Audit provides auditors, security compliance officers and audit administrators with analytical insight and transparency for identity and access. Based on historical identity data and recorded events from the identity and access management processes, DirX Audit allows answering the “what, when, where, who and why” questions of user access and entitlements. DirX Audit features historical views and reports on identity data, a graphical dashboard with drill-down into individual events, an analysis view for filtering, evaluating, correlating, and reviewing of identity-related events and job management for report generation.

For more information: support.dirx.solutions/about

Legal remarks

On the account of certain regional limitations of sales rights and service availability, we cannot guarantee that all products included in this document are available through the Eviden sales organization worldwide. Availability and packaging may vary by country and is subject to change without prior notice. Some/All of the features and products described herein may not be available locally. The information in this document contains general technical descriptions of specifications and options as well as standard and optional features which do not always have to be present in individual cases. Eviden reserves the right to modify the design, packaging, specifications and options described herein without prior notice. Please contact your local Eviden sales representative for the most current information. Note: Any technical data contained in this document may vary within defined tolerances. Original images always lose a certain amount of detail when reproduced.