EVIDEN

Identity and Access Management

DirX Directory

Release Notes

Version 9.1, Edition June 2025

Table of Contents

Release Notes

To run the DirX Directory Server you MUST have a LICENSE file and its corresponding signature file (see Chapter License Requirements)

To request your license file go to https://help.dirx.solutions Go to "DirX Support" and select "Request a License"

1. General

This Readme file contains important information about DirX Directory 9.1 (DirX) that are not described in the user documentation (Edition January 2025). Familiarity with the user documentation will make this readme file easier to understand.

1.1. Previous Releases

DirX Directory V8.9	July	2020
DirX Directory V8.10	March	2022
DirX Directory 9.0	November	2023

1.2. DirX Directory 9.1

See the datasheet provided on the DVD for highlights and features of DirX.

1.2.1. New in this Release

1.2.1.1. New Features

New functionality:

- Kubernetes deployment - From version 9.1 on, DirX Directory is also delivered in form of container images.The release contains also a Kubernetes example project which could be used to evaluate and experiment with the container images.For details, please check the new DirX Directory Containerization document.
- Code signing - verification of binaries: see Section 2.6 for details.
- License control - verification of DirX Directory license: see Section 1.2.8 for details.
- Time-based one-time password two-factor authentication (TOTP 2FA) for DirX Directory users: see chapter "14.8 Using Two-Factor Authentication (2FA)" in DirX Directory Administration Guide for details. Important note for shadowing configurations: TOTP 2FA should only be configured for a user, if all servers in a shadowing configuration support it (all servers are already updated to version 9.1).
- New dirxcp command "verify_abbr" was introduced to check the installed abbreviation files against the schema stored in the DSA. In connection with that new feature, the dirxabbr file has been synchronized to the default schema. Abbreviations that are not part of the default schema have been commented out and moved into a separate section labelled as "Historical Abbreviations". To reuse them if needed, move the selected abbreviations from this section to an application- specific dirxabbr-ext file and uncomment them there. See the section "Abbreviation Files" in the DirX Directory Administration Reference for details.
- New DSE type checks have been introduced in dbamverify, which upon detection of errors previously unseen, will block any new backups to be made. Therefore it is highly recommended to take a backup right before switching to the new version.

- Partial support for 4 bytes UTF-8 characters (emojis) in attributes with Directory String syntax was added. For details on limitations see Section 5.4.1 (DirX server limitations)
- Optimized import-dbconfig Tcl procedure for attribute index configuration The old version of this procedure processed all lines in DirXDBConfig.out index configuration file sequentially and issued one 'db attrconfig' command per 'attributeIndex: ...' line, creating all requested index types for this attribute. In the new version it combines up to 200 'attributeIndex: ...' lines for one index type (e.g: INITIAL, ...) into one single 'db attrconfig' command. This new approach is much faster, than the old method.

Diagnostics and logging:

- Software Bill of Materials file (DirX-Directory-sbom.json) containing information about delivered components and their versions was added to the DirX Directoty installation package.

1.2.1.2. Bug Fixes

DirX Directory 9.1 includes all bug fixes applied to the latest patch 9.6.678 of the former release version DirX Directory 9.0 plus the following fixes.

- SDX-1039: Non RFC compliant Server Side Sorting The Server Side Sorting implementation did not behave as described in RFC-2891#section-2, point 4. It specifies that if the criticality is false and the server "for some reason cannot sort the search results", then "the server should return all search results unsorted". This is fixed now. If you want to make sure that the result is sorted, set the criticality to true.
- SDX-1010: New restriction was introduced in the DSA for search filters A huge number of filter items can significantly increase the memory consumption of a DSA during the evaluation of the filter in a search operation. This could lead to consuming the whole available CTX memory, causing problems in other parallel running operations (e.g.: DISP update) as well. To avoid this situation, a limit was introduced in the DSA (controlled via environment variable DIRX_SEARCH_FILTER_ITEM_LIMIT) that sets the max upper limit for filter items contained in a search operation. If this environment variable is not set, the default limit value of 5000 filter items is used.
- SDX-1006: DSA crash on index configuration A software bug in libdbam could result in a DSA crash when a db attrconfig command was executed and the database had more than 800 configured indexes.
- SDX-991 userPassword attribute added with dirxadm visible in DSA logs in cleartext The command line of modify commands executed with dirxadm are logged in ADM*, DSA_EXC* and audit logs. Therefore if the UP attribute was added with dirxadm, the PW value was visible in clear text in the logged command line. With this modification the behavior has changed, the added PW value will be replaced with '**' in the logged command line.
- SDX-965: Incorrect output in lob show -pretty mode Software bug in dirxadm. The lob show command in -pretty mode produced incorrect output, every line was left aligned. Therefore a complex specification filter containing several And/Or/Not items was incorrectly displayed. Fix: Use correct indentation according to the displayed structure.
- internal: notes for dbamverify - related to automatic database verification In this patch version, the handling of binary backup header was improved. If the dbamverify is called

with a subset of the components, the verification result components will not be reset, only the newly verified components will be added in addition. This means, that it is suported now to do the backup verification in individual steps. For example, the following sequence:

- Save the backup without verification using dirxbackup -n -S backup
- Verify the D, S and T components by using dbamverify -DST backup
- Do the rest of the verification by using dbamverify -AX backup This sequence will result in a fully verified backup file.

1.2.1.3. Discontinued Features

The dirxbackup -T option is removed. It was used in the past for doing only basic tests on binary backups. This confused users as the backup could contain errors while the -T option returned with "archive ok". As the dbamverify is used now for doing a detailed verification of binary backups, this option is removed.

The LDAP Mib interface in dirxadm is deprecated. It will not be supported in future versions of DirX Directory. Use of the LDAP extended operations is recommended starting from version DirX Directory 8.4.

1.2.1.4. Changes to the User Interface or Configuration

No changes since DirX Directory 9.0. No new SNMP traps since DirX Directory 9.0.

1.2.2. Supported Platforms

DirX Directory 9.1 is available as a native 64-bit application on the following platforms:

System Family	Operating System
Intel	Windows Server 2019
Intel	Windows Server 2022
Intel	Red Hat Linux ES 8 (Tested up to 8.10)
Intel	Red Hat Linux ES 9 (Tested up to 9.5)
Intel	SuSE Linux ES 12 (Tested up to 12 SP5)
Intel	SuSE Linux ES 15 (Tested up to 15 SP5)
Intel	VMware ESXi with guest operating systems listed above that are supported by VMware ESXi
Intel	Kubernetes (minikube) with the host operating systems listed above

See also section 1.2.7 titled "Software Requirements" for details about the Linux platforms.

The product is tested with all supported operating systems on VMs running on VMware ESXi6.0, ESXi6.5 and ESXi6.7 hosts.

1.2.3. Delivery Packages

This section provides information about DirX Directory 9.1 delivery packages on the supported platforms.

The following software packages are available:

Name	Description
DirX-SV V9.1	X.500 Directory System, Server package [1] incl. DSA with database system and LDAP Server
DirX-CL V9.1	X.500 Directory System, Client package [2] incl. LDAP Server
DirX Manager	Graphical administration interface [3]

(1) On Linux platforms, the Server package is combined into a tar file.

(2) The client package is not available on Linux platforms.

(3) All programs are native 64-bit applications.

1.2.4. Distribution Media

Software packages for all platforms are distributed on the following DVD: DirX Directory 9.1 (Edition 01/25)

In addition to the distribution medium, you must purchase separate product licenses in order to use the software packages.

Please contact your local sales representative for details on product licenses.

1.2.5. User Documentation

This section provides information about DirX user documentation.

1.2.5.1. DirX User Manuals

Title	File Name
DirX Directory Introduction	DirX_Directory_Introduction.pdf
Administration Guide	DirX_Directory_Administration_Guide.pdf
Disc Dimensioning Guide	DirX_Directory_Disc_Dimensioning_Guide.pdf
Guide for CSP Administrators	DirX_Directory_Guide_for_CSP_Administrators.pdf
Administration Reference	DirX_Directory_Administration_Reference.pdf

Title	File Name
Syntaxes and Attributes	DirX_Directory_Syntaxes_and_Attributes.pdf
LDAP Extended Operations	DirX_Directory_LDAP_Extended_Operations.pdf
External Authentication	DirX_Directory_External_Authentication.pdf
Supervisor	DirX_Directory_Supervisor.pdf
Plugins for Nagios	DirX_Directory_Plugins_for_Nagios.pdf
DirX LDAP Proxy	DirX_Directory_LDAP_Proxy.pdf
Manager Guide	DirX_Directory_Manager_Guide.pdf
Best Practices for DB Error Recovery	DirX_Directory_Recovery_Best_Practices.pdf
Containerization	DirX_Directory_Containerization.pdf

The edition of all manuals is January 2025.

The files are located on the DVD under the folder Documentation/DirXDirectory.

You need Adobe Acrobat Reader 7.0 or newer to view the PDF files. For a free copy of Adobe Acrobat Reader please refer to

http://get.adobe.com/reader/

or to

http://www.adobe.com

1.2.5.2. Tcl V8.3 Commands

The online documentation set includes the reference pages of the Tcl V8.3 commands. Please refer to the file Documentation/tcl_V83_part/license_terms.txt for license agreements.

1.2.6. Hardware Requirements

This section provides information about hardware requirements.

1.2.6.1. RAM

At least 8 GB RAM is required for DirX servers.

1.2.6.2. Disk Space

For the default configuration, you need 3 GB for installation data of DirX.

Operation of DirX requires disk space for log files, LDIF files, audit files, journal files and other temporary files for post indexing, database verification and other purposes. Consider at least additionally 20 GB disk space for these files.

For calculating the required disk space for the DBAM database, you should look into the Disc Dimensioning Guide.

1.2.6.3. Virtual Machines

When running DirX Directory on a virtual hardware, it is essential for a stable and performant service that the resources assigned to the guest system are available at any time.

This applies to the assigned CPUs, the I/O throughput to persistent storage and especially to the assigned RAM, as DirX Directory uses it for its DBAM Cache and requires a performant access to the main memory.

Additionally care must be taken to use up-to-date and correctly configured driver software implementing the VM's network interface.

Thus it is essential to design the virtual machine properly and perform tests to assure that the expected throughput can be achieved. Special considerations must be taken with respect to the number of cores assigned to the VM relative to the number of cores that are physically on the host system.

We highly recommend to install DirX Directory on a separate VM without any further major services running aside.

1.2.7. Software Requirements

DirX requires the presence of one of the supported operating systems.

On Windows, NTFS (not FAT) must be used.

On Linux, DirX is installed in the home directory of a user id. The shell interpreter "bash" must be installed.

TCP/IP must be configured and running. This is a requirement even if you work with a stand-alone system where all directory applications and the directory server run locally on your system. A DNS service must be running that allows hostname to IP address translation for all involved hosts.

DirX Directory runs by default under the control of the systemd. In case [x]inetd shall be used, this package must be installed manually. Alternatively the DirX Service may be started from a shell. See section 6.8 in this document for details.

On Linux platforms, gzip is required to be installed by default. The minimum version required is gzip 1.3.5. The installed gzip version is displayed by the command gzip -V. On Windows platforms, DirX ships a gzip version and installs it in the bin folder of DirX installation path.

On Red Hat Linux version 8, if the security module SELinux is used in "enforcing" mode, the SELinux policies for the DirX product must be properly configured immediately after the installation. Check the output of the "sestatus" command to see the mode of SELinux. If the SELinux mode is "disabled" or "permissive", no special SELinux configuration is necessary

for DirX. In case of "enforcing" mode, a default configuration can be done by executing the following command:

```
$ sudo
<DIRX_INST_PATH>/scripts/selinux/configure_dirx_selinux_label.sh
<DIRX_INST_PATH>
```

This command configures the DirX product to run within the so-called "unconfined" domain. For more information on SELinux please refer to the Red Hat Linux ES 8 documentation. Note that SELinux in "enforcing" mode is supported on RHEL8 with DirX running in the "unconfined" domain only.

1.2.7.1. Packages to be installed manually on LINUX

On Linux, the following preconditions apply:

Mandatory packages: For the DirX Directory application, the 64-bit version of the following packages are mandatory required: glibc, libgcc, libstdc++, libuuid, zlib

Recommended packages: In order to be able to evaluate possible crashes on Linux, the DirX crashlibrary tool needs the existence of the gdb and gstack tools. Therefore the installation of the GNU debugger package (the most recent release of GDB) is strongly recommended.

Optional packages: The following packages are optionally required for a DirX Directory installation: optionally xinetd

1.2.7.2. Local Port range to be checked on Linux

On Linux, the setting of the port range for local ports (ephemeral) should be:

net.ipv4.ip_local_port_range = 32768 61000

This setting prevents that the system assigns a dynamic port that is also configured as a listen port.

Red Hat 7 and later:

Edit the /etc/sysctl.conf file and add the following line:

```
# Allowed local port range
net.ipv4.ip_local_port_range = 32768 61000
```

You must restart your network for the change to take effect.The command to manually restart the network is:

```
[root@deep] /# /etc/rc.d/init.d/network restart
```

1.2.8. License Requirements

This version of DirX Directory introduces license control for the DirX Directory Service.With this feature:

- The license terms for using a DirX Directory service installation are defined in a configuration file supplied by the DirX Directory vendor and installed with the product.
- At every service startup and daily on a running system, the service checks the settings in the file against the current installation for possible license violations and verifies that the license configuration file has not become corrupted.
- The service logs the results of the check and takes additional actions depending on the "license type" setting in the license configuration file.

DirX Directory's license control feature provides an easy way for you (the customer) to detect expired licenses and/or license limitations that have been exceeded in your installation. The next sections provide more details about this feature.

1.2.8.1. License Files

DirX Directory license control requires two files to be present in the $DIRX_INST_PATH/conf folder:

dirx.lic

a human-readable text file that uses a set of parameters to specify the terms and conditions for use of the DirX Directory product. The DirX DSA's license validation procedure compares the settings in this file with the current installation to verify that it is in compliance with the license terms and conditions. For a description of dirx.lic file format and parameters, see the section "DirX Directory License Files" in the "DirX Directory Files" chapter of the DirX Directory Administration Reference.

dirx.lic.sig

a binary signature file generated from the dirx.lic file. The DSA's license validation procedure compares this file to the dirx.lic file in use by the service to ensure that it has not been tampered with or corrupted. For details about code signing and verification, see section 2.6 and the section "Code Signing Files" in the "DirX Directory Files" chapter of the DirX Directory Administration Reference.

Although dirx.lic is a simple text file, DO NOT CHANGE IT. Changing the file will cause code validation of dirx.lic against dirx.lic.sig to fail and your service will not start! See Section 2.6 for further details about this process.

The dirx.lic and dirx.lic.sig files are not part of DirX Directory's regular backup procedure. You must back up these files manually.

1.2.8.2. License Types

DirX Directory license control recognizes two license types: a "**trial**" license and a "**perpetual**" license.

The license type is recorded in a parameter in the dirx.lic file and is used during license checking to determine how to handle a license violation. See section 1.2.8.5 and the section "DirX Directory License Files" in the chapter "DirX Directory Files" in the DirX Directory Administration Reference for further details.

1.2.8.2.1. Trial License

The default dirx.lic file delivered with each installation is configured as a trial license. A trial license has the following characteristics:

- Allows a maximum of 2000 directory entries (sufficient for running the My-Company sample database for demonstration purposes)
- Can be used for an unlimited amount of time
- Is only valid for a specific product version and thus cannot be used on any newer product version
- Does not support opening trouble tickets

A DirX Directory upgrade installation installs the most recent trial dirx.lic and dirx.lic.sig files into $DIRX_INST_PATH/conf unless it finds that these files are already present. In this case, it installs the most recent trial license files in the same directory as the existing license files with the names dirx.lic.new and dirx.lic.sig.new.

1.2.8.2.2. Perpetual License

A "perpetual" license is a dirx.lic file with parameter settings configured to represent the terms and conditions negotiated between you (the customer) and the DirX Directory vendor for use of the DirX Directory product. If you have a larger DBAM database or initially need more than 2000 entries, or you have other specific requirements, you need to obtain a perpetual license from the DirX Directory vendor.

A perpetual license allows both you and the vendor to limit DirX Directory usage to specific environments: for example, by allowing up to two million directory entries but restricting DirX Directory service operation to specific hostnames or IP addresses.

When requesting a perpetual license, you will be asked about restrictions. You can also bring your own requirements, which in turn may influence the price and capabilities of maintenance support.

A perpetual license obtained for a major release of DirX Directory (for example, DirX Directory 9.x) is valid for minor releases (for example, 9.1, 9.2, etc.) and patches. This means you can re-use your existing perpetual license on minor version and patch releases.

1.2.8.3. Obtaining a New License

To obtain a new license, contact Eviden DirX sales or send your request using the DirX

Support portal https://help.dirx.solutions/.

If you already own a perpetual license and just need new dirx.lic and signature files for your installation, send a request using the DirX Support portal https://help.dirx.solutions/.

1.2.8.4. Installing a New License

Once you receive your new license and signature files from the DirX Directory vendor, you can simply overwrite the existing dirx.lic and dirx.lic.sig files in your installation's /conf folder and then restart the service. The license check performed at startup automatically updates the service with the information from the new dirx.lic file.

Alternatively, you can leave the service running and wait for the next automatic license check to update the service, or you can use the dirxextop LDAP extended operation dsa_license_check to trigger the license check manually. For details, see Section 1.2.8.5.3.

1.2.8.5. License Validation Procedure

License validation is part of the DirX DSA process. At product installation, at each service startup, and at least once a day on a running system, the DirX DSA's license validation procedure:

- Compares the signature file dirx.lic.sig with the dirx.lic file in the installation.
- Compares the parameter settings in the dirx.lic file against the current installation and logs the results in the DSA's fatal log file ($DIRX_INST_PATH/server/log/fatalDSA*).

If the signature file does not match the dirx.lic file, the DSA shuts down or does not start.

If a license violation is detected and the license is a trial, the DSA shuts down or does not start.

If a license violation is detected and the license is perpetual, the actions taken by the service depend on the type of violation:

- If the violation is because vendor support has expired or because the installation has exceeded the maximum number of directory entries, the DSA continues to run or starts but generates warning messages in the DSA's fatal log file. For example, if your license allows 100000 entries but your installation has 120000, the service still starts but you will find WARNING messages in the DSA's fatal log file about the problem and you may not be able to open trouble tickets if your license settings are not met.
- For all other types of violation, the DSA shuts down or does not start.

The license validation procedure also updates the DSA with the settings found in the dirx.lic file.

1.2.8.5.1. Adjusting Daily License Checking

By default, the license validation procedure runs automatically every 86400 seconds (1 day) after DSA start until the DSA is stopped. You can increase the frequency of daily license checking by setting the environment variable:

```
DIRX_LICENSE_CHECK_INTERVAL=xxxxx
```

where xxxxx is the number of seconds between the checks. Allowed values are in the range [300-86400]. See the chapter "Environment Variables" in the DirX Directory Administration Reference for details on environment variables.

1.2.8.5.2. Viewing Current DSA License Settings

You can view the license information currently being used by a running DSA by specifying the dsa_license_info LDAP extended operation (OID 1.3.12.2.1107.1.3.2.13.50) to the dirxextop command. For details, see the description of the dirxextop command in the DirX Directory Administration Reference.

1.2.8.5.3. Triggering a License Check

You can use the dsa_license_check LDAP extended operation (OID 1.3.12.2.1107.1.3.2.13.51) to the dirxextop command to perform an on-demand license check procedure.For details, see the description of the dirxextop command in the DirX Directory Administration Reference.

2. Installation

This section provides information on how to install DirX.

2.1. Initial Installation

The initial installation prepares networking over IDM stack.

2.1.1. Installation Procedure on Windows Platforms

On the English version of Windows systems the default base directory is "C:\Program Files\DirX\Directory" for the 64-bit version of DirX.On Windows systems with other language packages installed than English the default base directory may differ. The base directory for installation is under administrator control: The administrator can choose a different pathname.

It is strongly recommended to run the DirX Service under the administrator's account. If the DirX Service runs under the system account it is not possible to perform a dirxbackup command while the service is running.

The execution of the dbam command line tools dbamconfig and dbamboot and the backup tool dirxbackup require to be executed from a cmd shell that was started in the "run as administrator" mode.

Steps of the installation procedure :

a. Log in as administrator + disable UAC
b. Install DirX

Insert the DVD. The installation procedure starts automatically. A menu is offered to choose the product.

Choose the DirX Directory product.

Choose from the options offered during the installation steps.

Note that the service must run under an account with administrator rights.

Only the 64-bit version of DirX is available

After the installation procedure on Windows, 4 logfiles can be found on the system:

- the InstallAnywhere logging named DirX_Directory_<version>_Install_<date_time>.log located in <dirx-inst-path>/tmp
- the InstallAnywhere logging named dirxdirectory_installer_debug.txt located in <dirx-inst-path>/logs after successful installation, and in <user_temporary_folder> (C:\Users\<user>\AppData\Local\Temp\) written during first steps of installation
- the application log file named dirxdirectory_debug.txt located in <system-temp> (c:\tmp\)

In the latter two files log entries are appended preserving older installation debug logging.

2.1.2. Installation Procedure on Linux Platforms

DirX 9.1 has to run under a normal user id, the DirX account.

a. Log in as superuser

b. Mount the DVD on a directory:
mount <dvd-device> <mount-point>
e.g.
`mount /dev/scd0 /mnt`

c. Log in under the DirX account

d. Extract the tar archive to a temporary directory, e.g. `/usr/tmp/dirx`

```
mkdir /usr/tmp/dirx
cd /usr/tmp/dirx
tar xvf /mnt/Installation/Linux/DirXDirectory/DirXServer64/dirx*
```

tar creates temporary subdirectories containing all files for DirX-SV and an installation script.

At least 25 MB of free disk space must be available in the file system where the temporary directory resides. After installation, you can remove the subdirectory dirx.

e. Install DirX with the script dirxinst

1.
```
cd /usr/tmp/dirx
./dirxinst /usr/tmp/dirx <inst-directory>
```

where the installation folder <inst-directory> may be the HOME folder of the DirX account or a special dedicated installation folder owned by the DirX account performing the script.

2. Log in as superuser
3.
```
cd /usr/tmp/dirx
bash ./dirxinst_root /usr/tmp/dirx <HOME folder of the DirX
account>
```

The second argument must be the HOME folder of the DirX account.

This script changes access permissions on executable files and changes xinetd configuration data.

The scripts also assume that Bash is the login shell of the DirX account. dirxinst inserts therefore the call to .dirxrc into the $HOME/.bash_profile.
In case the Bash is not the login shell, the call to .dirxrc is inserted into $HOME/.profile.

Without execution of .dirxrc during the login procedure pathnames will not be set correctly. It also adds settings to the user's .bash_profile.

4. Again log in under the DirX account. (Due to the .dirxrc call in your .bash_profile, some important variables are set when performing the login process.)

f. Remove the temporary tar extract

```
rm -r /usr/tmp/dirx
```

g. Automatic Start and Stop of DirX

1. If the system does not support insserv(8):

   If you want the DSA and the LDAP server to stop automatically with every system shutdown, you must create the following link:

   `ln -s <installation base directory>/etc/dirx /etc/rc.d/rc0.d/K49dirx`

   If you want the DSA and the LDAP server to start automatically with every system start, you must create a link in the appropriate /etc/rc.d/rc<runlevel>.d directory.

   Example:

   To start DirX during runlevel 3 create the following link:

   `ln -s <installation base directory>/etc/dirx /etc/rc.d/rc3.d/S99dirx`

2. System supports insserv(8), e.g. SuSE ES 12:

   Copy the DirX start/stop script to /etc/init.d and use insserv to enable automatic starting and stopping of the DirX service. Perform the following commands:

   ```
   cp <dirx-inst-path>/etc/dirx /etc/init.d
   insserv dirx
   ```

After the installation procedure on Linux, a logfile can be found on the system:

- the installation log file is stored in <inst-path>/tmp and is named dirx_install_<date>_<number>.log.

Each installation creates a new installation log file, the _<number> part of the name is used only if a file with that name already exists.

2.1.2.1. Use of systemd on Linux

The systemd is the default service used to control the DirX Directory service.

The systemd is a daemon (process 1, formerly init) to start, monitor and finish system daemon processes. It therefore replaces the SysV init system and xinetd besides providing other functions. Systemd was adopted in 2014 by the Redhat Enterprise Linux 7 and SUSE Linux Enterprise Server 12 distributions. Other distributors, like Debian, openSUSE, Fedora and Ubuntu, had used systemd already some years before.

DirX installations before 8.7 required the systemd-SysV-compatibility mode during boot of the system to start DirX and stop it at shutdown. The file <DirXaccount>/etc/dirx had to be installed manually by the Administrator in /etc/init.d and he had to create symbolic links in /etc/rc?.d with a convenient name S???dirx resp. K???dirx for the start and kill scripts. This is not necessary any more since DirX 8.7 .

Instead, the DirX installation comes with a template file <DirXaccount>/etc/dirx.service, that is adjusted to the DirX Account by the dirxinst script (replacing the string DIRINST by the actual path to the DirX installation folder) similar to the file <DirXaccount>/etc/dirx . The file <DirXaccount>/etc/dirx ist called from the etc/dirx.service file with the parameters "start" or "stop" through the ExecStart and ExecStop directives in the [Service] section similarly to the former SysV mechanism. The dirxinst_root script copies the dirx.service unit file to the folder /usr/lib/systemd/system and prints a notice, in case an old DirX start script is found in /etc/init.d, which should be removed together with its symbolic links from the rcX.d folders.

The next change for DirX is the replacement of the xinetd port listening strategy by the systemd functionality. For this purpose, dirxinst_root creates two files in /usr/lib/systemd/system: dirx-listen.socket and dirx-listen@.service. They contain the necessary information about the listening port, the DirX Account and the start script, which were formerly found in /etc/xinetd.d/dirx.

Then systemd is updated and the DirX listening and boot service are enabled and started:

```
# systemctl daemon-reload
# systemctl enable dirx-listen.socket
# systemctl start dirx-listen.socket
# systemctl enable dirx.service
# systemctl start dirx.service
```

The systemd status can be displayed with "systemctl status" or "systemctl status dirx-listen.socket" for the dirx-listen.socket only. Failed services are displayed with "systemctl --failed". Further information about a failed service with process-ID <pid>, which is found in the status information, can be displayed with "journalctl _PID=<pid>".

To deinstall the DirX configuration for systemd, you have to run dirxdeinst as root.

During an update installation on systems using systemd, DirX will now switch from using xinetd to systemd. The dirx configuration file is moved

from /etc/xinetd.d to <DirXaccount>/tmp and signal SIGHUP is sent to the xinetd to reread the configuration. The system administrator must then remove the xinetd dependency in the header line "# Required-Start: " of the customer created S???dirx or K???dirx scripts. The dirxinst_root script will print a notice about the special administrative tasks, when it founds an xinetd DirX configuration or a SysV start script in /etc/init.d in a systemd environment. A new DirX installation creates the proper template <DirXaccount>/etc/dirx when dirxinst is called.

The scripts dirxinst_2nd and dirxint_2nd_root for parallel DirX installations avoid collisions in service unit file names by prefixing them with the alternate DirX account name.

2.1.2.2. Change of the default service port

DirX uses by default the port 5800 to communicate with the service start demon process of the system. In case this portnumber is already in use, you may change the value by editing the script dirxinst_root. Replace the portnuber 5800 in the assignment DIRX_DEF_PORT=5800 by the desired port number value.

Note that you will have to supply the specific portnumber in the dirxadm start command sys start -port <specific-port-number> in order to start the DirX service on the local host.

2.1.2.3. Enable schema LDAP name checking according to standard

Strict LDAP name checking is disabled by default for compatibility reasons with older versions. In a new installation environment (with empty database) the flag "set DIRX_DSA_ENABLE_SCHEMA_STRICT_NAME_CHECK=1" should be added to dirxenv.ini in order to ensure better compliance with RFC. With this setting invalid attribute or object class names will be rejected by DSA. Name syntax is described in "Syntaxes and Attributes" "2.7.1 Attribute-Type-Description" and "2.7.2 Object-Class-Description".

2.1.3. Second DirX-installation on a LINUX-Server

In DirX 9.1 installation and deinstallation scripts are provided that allow to install and run a second instance of the Dirx Directory service on one Linux host.

2.1.3.1. Prerequistes and general Remarks:

0. systemd and/or xinetd is up and running
1. Each such DirX installation has to be performed under a different LINUX user account only
2. Each such DirX installation (except the very first one) has to be configured separately after unzipping the DirX package to a temporary folder. This configuration task comprises in separating the various port settings.
3. Don’t setup log, audit or even conf folders to a common shared disk folder. This means, each DirX installation has to use of its own log, audit and conf folders.
4. Keep in mind that the number of sockets is limited to 64K per LINUX server instance.

Thus, it is not recommended to establish more than two DirX installations per HW server.

It is assumed, there is one DirX installation performed in default mode under a certain user account (e.g. dirx01). Now, a second DirX installion is planned to be established under the new user-account (e.g. dirx02). To summarize all based on user accounts:

user=dirx01:	default DirX installation while applying the common installation scripts dirxinst and dirxinst_root (see $HOME)
user=dirx02:	second DirX installation to be configured first while applying the variant of the scripts for the second installation, i.e. dirxinst_2nd and dirxinst_2nd_root.

The second DirX user-account must not be "dirx" as this name is internally reserved for the first (default) DirX installation already.

2.1.3.2. Preparing the second DirX installation

These steps have to be performed as user=dirx02

1. Create a new folder under $HOME/temp (e.g. /home/dirx02/temp) mkdir $HOME/temp → userName=dirx02
2. Extract the DirX-V9x package in the new $HOME/temp folder. gzip -dc dirx90_9.6.xxx.ddmm.x86_lx-64.tar.gz | tar xf -
3. Adapt dirxinst_2nd for your needs, as follows:

2.1.3.3. Performing the second DirX installation

These steps have to be performed as user=dirx02

0. Passed parameters:

   ```
   $1: pathname of the unpacked DirX package
   $2: DirX installation pathname
   ```

1. PSAP port number:

   The default port in the DSAs PSAP is set to 22200 in the second DirX installation. If you wish to use another portnumber, change the value in the following statement in dirxinst_2nd

   ```
   vcomplete[v_NoElem]="DIRX_OWN_PSAP=\"TS=DSA1,NA='$SPEC_COM'TCP/IP_
   IDM!internet='$DIRX_LOCAL_IP'+port=22200'$SPEC_COM'\"";
   ```

2. DSA PMAP port number:

   The default port for the DSA RPC portmapper is set to 8999 in the second DirX

installation. If you wish to use another portnumber, change the value in the following statement in dirxinst_2nd

```
vcomplete[v_NoElem]="DIRX_PMAP_PORT=8999
```

3. LDAP PMAP port number:

   The default port for the LDAP RPC portmapper is set to 9999 in the second DirX installation. If you wish to use another portnumber, change the value in the following statement in dirxinst_2nd

   ```
   vcomplete[v_NoElem]="DIRX_LDAP_PMAP_PORT=9999
   ```

4. DSA name

   The default DN for the DSA is set to "CN=${userName}-DSA-${DIRX_HOST_NAME}", where $userName expands to the Linux account name and ${DIRX_HOST_NAME} to the hostname of the machine. If you wish to use another name, change the value in the following statement in dirxinst_2nd

   ```
   DIRX_DSA_NAME="CN=${userName}-DSA-${DIRX_HOST_NAME}"
   ```

2.1.3.4. Completing the second DirX installation

These steps have to be performed as user=root

0. Passed parameters:

   ```
   $1: pathname of the unpacked DirX package
   $2: DirX installation pathname
   ```

1. Assignments made in dirxinst_2nd_root

   The script dirxinst_2nd_root sets xinetd listening port to 15800 by means of the following assignment:

   ```
   DIRX_DEF_PORT=15800    ... listening for the systen
   ```

   Automatically assigned in dirxinst_rootV90:

   ```
   SYSTEMD_DIR=/usr/lib/systemd/system
   SYSTEMD_LISTEN=${userName}-listen
   SYSTEMD_LISTEN_SOCKET=$SYSTEMD_DIR/${SYSTEMD_LISTEN}.socket
   ```

```
SYSTEMD_LISTEN_SERVICE=$SYSTEMD_DIR/${SYSTEMD_LISTEN}@.service
```

Automatically added to /etc/services:

```
${userName}   $DIRX_DEF_PORT/tcp    # DirX servive for user
${userName}
```

Automatically created in $SYSTEMD_DIR (default-file: dirx-listen.socket)

```
New file name:  $SYSTEMD_LISTEN_SOCKET with follwoing content:
-
[Unit]
Description=DirX listening socket

[Socket]
ListenStream=$DIRX_DEF_PORT
Accept=yes

[Install]
WantedBy=sockets.target
-
```

Automatically created in $SYSTEMD_DIR (default-file: dirx-listen@.service)

```
New file name:  $SYSTEMD_LISTEN_SERVICE with follwoing content:
-
[Unit]
Description=DirX start listen service

[Service]
ExecStart=/home/${userName}/bin/dirxdsa.sh
User=${userName}
StandardInput=socket
-
```

Automatically processed:

```
systemctl daemon-reload
systemctl enable ${SYSTEMD_LISTEN}.socket
```

```
systemctl start  ${SYSTEMD_LISTEN}.socket
```

2.1.3.5. Post second installation tasks

1. Make the settings in $HOME/.dirxrc effective

```
. $HOME/.dirxrc  (or while relogin with: su - dirx02)
```

2. Port settings (in particular the DAP port settings in config files)

   Assign the same DAP-Port (as assigned before in C-1) in:

```
<inst-path>/client/conf/dirxcl.cfg and <inst-
path>/ldap/conf/dirxldapv3.cfg
```

   e.g. like: `/CN=DirX-DSA TS=DSA1,NA='TCP/IP_IDM!internet=127.0.0.1+port=22200'`

3. ldap configuration subentries

   Prior to startup the DirX-Service with dirxadm make sure the ldapport numbers (plain ldap as well as secure ldap port) may by distinct to those ones used in DirX service-1 (user=dirx01) otherwise the DirX-Service-2 (user=dirx02) will refuse to startup. In this case, first start the dirxdsa process manually and continue modifying the respective ldap numbers accordingly, i.e. modify the respective ldapconfgurations via DAP by replacing the attributes LPNU and LSPN with the values of the desired ldap ports.

   

   In terms of a floating master configuration on the same HW-Server, at least two ldapconfiguration subentries now will have to be established, one for the DB-Master (e.g. user=dirx01) and one for the DB- Consumer (e.g. user=dirx02) in order to apply distinct ldap port numbers for both DirX services running on the same HW-server. Use the environment variable DIRX_DEFAULT_LDAP_SERVER to define the ldapconfiguration name referring to the alternate default ldapserver port for the second DirX installation. In case of more than 1 ldapserver per installation specify values for the DIRX_ADDITIONAL_LDAP_SERVERS environment variable in each DirX installation.

4. Automatic DirX-Startup on server reboot
   1. First DirX-Installation
      a. Follow the instructions in the ReleaseNotes
   2. Second DirX-Installation
      a. Copy $HOME/etc/dirx to $HOME/etc/<userName>
      b. Edit $HOME/etc/<userName> while applying the same port number (DIRX_DEF_PORT - see dirxinst_rootV9x) to the dirxadm-start command

c. Follow the instructions in the ReleaseNotes while applying the new $HOME/etc/<userName>

2.1.3.6. Deinstalling the first and the second DirX installation

In order to remove the second DirX installation apply the script dirxdeinst_2nd running in root-user mode after stopping the DirX service in scope.

If one wants to remove the first (user=dirx01 here) DirX installation, also stop the DirX service of user=dirx02 and apply the dirxdeinst script running in root- user mode

In case of a single DirX installation apply the common dirxdeinst script instead running in root-user mode.

During deinstallation all files and folders below <dirx-inst-path> which were created by the install script will be deleted.This includes all private files and folders in these install script created subfolders. Private files/folders added directly under <dirx-inst-path> will be preserved during deinstallation.

2.2. Upgrade Installation

An upgrade installation does not clean up the existing installation folders. The following existing configuration files will not be overwritten:

- dirxcl.cfg
- dirxldap.cfg
- ldap/conf/dirx_pkcs12.pwd
- ldap/conf/cert_ldapserver.p12
- ldap/conf/cert_ldapserver.der
- ldap/conf/testCA.der
- client/conf/cert8.db
- client/conf/key3.db
- conf/snmptraps.cfg
- conf/dirx.lic
- conf/dirx.lic.sig
- monitoring/supervisor/setup_common_data.tcl
- http/conf/dirxhttp.cfg
- http/conf/http.pem
- http/conf/http_pkcs12.pwd

The dirxabbr file format has not changed, but some new elements may have been added. The dirxabbr file of the old DirX version is renamed to "dirxabbr.old".

Create application-specific dirxabbr-ext files with all your project specific extensions. You

avoid to re-enter your project-specific extensions to the newly installed dirxabbr file when you perform an upgrade installation in the future. See section "Abbreviation Files" in the DirX Administration Reference for details.

All other existing files will be overwritten when they also exist on the installation DVD, for example dirxlog.cfg.

Before you start the upgrade processes it is recommended to save the existing database with dirxbackup, and also export it to LDIF files.

The format of DSA audit files and LDAP audit files has been changed. Before performing the DirX upgrade installation you should evaluate the binary audit files with the dirxauddecode or dirxaudstatistics tool of the predecessor version. Before starting DirX service after upgrade installation the audit.log files from DSA and LDAP server should be removed since their format is incompatible and the DirX service might not start.

DirX V8.10 introduced a stricter LDAP name check in schema (switched off by default). This verification might be turned on in future releases. In order to prevent any problems with future releases, this is a good opportunity to test whether your existing DB complies to the rules.

Steps to do this verification:

step 1

Start a new terminal window

step 2

Set environment variable:
"set DIRX_DSA_ENABLE_SCHEMA_STRICT_NAME_CHECK=1"

step 3

Execute "dbamverify -D" on the online DB or on a backup. This will display a list of found invalid names, if any.

step 4

Rename the listed invalid attribute or object class names to be standard compliant.

The default schema in DirX Directory 9.1 has been extended by object classes and attribute types used for functional or operational purposes. After a successfully upgrade installation the DSA of DirX Directory 9.1 generates all these attributes automatically if they are not found in the schema of an existing database. The same applies to the change from single to multivalued attribute types and to extensions of the affected object classes.

Upgrading DirX DSAs within a shadowing environment takes much more attention then an upgrade of a standalone DirX DSA. Please read carefully chapter "2.4 Upgrade Installation in a Shadowing Environment" in this document.

The LDAP controls supported by the current version of the DirX DSA are stored in the SCON attribute of the /CN=ldapRoot element of the database. This information will only be added at database creation time (dbamboot), and will not be updated automatically during a later upgrade installation. If the database was created with an older version of the DirX DSA, the

list of supported LDAP controls must be updated manually via dirxadm after the upgrade installation. From DirX Directory V8.9 to V8.10 the following LDAP controls were added:

1.3.12.2.1107.1.3.2.12.12 - LDAP Increment Replace Control

1.3.12.2.1107.1.3.2.12.13 - LDAP Matched Value Only Filter Control

New DSE type checks have been introduced in dbamverify, which upon detection of errors previously unseen, will block any new backups to be made. Therefore it is highly recommended to take a backup right before switching to the new version.

2.2.1. Upgrade Installation on Windows

It is required to close the Windows Event Viewer and the Windows Service Manager before starting the DirX upgrade installation on Windows.

To upgrade your previous installation install DirX Directory 9.1 from your DVD as described in section titled "Initial Installation" above. The installation procedure performs an automatic data migration.

2.2.2. Upgrade Installation on Linux Platforms

In order to perform a software upgrade from DirX Directory 8.9 or older to DirX Directory 9.1, the executable dirxdsas has to be deleted manually. The file dirxdsas is located in <dirx-inst-path>/bin.

Afterward you can follow the instructions given in 2.1.2 to overwrite the existing installation with the new release.

2.3. Upgrade installation and database migration

2.3.1. Upgrade Installation from DirX V8.2B and later versions

The DBAM database structure of DirX 8.2B and later versions is fully upward compatible with that of DirX 9.1. An upgrade installation is possible without performing any DBAM database conversion procedures.However due to new features and functional changes in DirX 9.1 additional migration steps are required as described later in this chapter.

2.3.2. Migration of ldapserver’s own keymaterial

Starting with DirX Directory 8.2B the OpenSSL libraries in the ldapserver are used in order to support LDAP protocol over SSL/TLS. There is no migration of ldapserver keymaterial neccessary in this case.

It is required that the content of a PEM File with the ldapservers private key and public key certificate is stored in the attribute ldapOwnKeymaterialPEM. It is still possible to have the key material stored in a PEM file on disk referenced by the attribute ldapOwnKeymaterialFile. The LDAPserver will prefer the content from ldapOwnKeymaterialPEM but if the attribute is missing it will try to read the PEM from file.

DirX 9.1 will install sample-keymaterial, which is not intended to be used in a real-life scenario. If old PKCS#12 key material exists that shall be used for the new DirX version, it has to be converted from the PKCS#12 format to the PEM format and stored in the attribute ldapOwnKeymaterialPEM (or referenced by ldapOwnKeymaterialFile). We recommend to use storing it to the subentry as this has the advantage that it automatically gets included when generating DirX DB backups.

In order to perform this conversion, the DirX Directory installation contains the command line tool 'openssl' from the OpenSSL package. Even if OpenSSL is already installed on your host, we recommend to use the 'openssl' tool from our package. With this tool the conversion can be done in the following steps:

If old PKCS#12 was stored in SSL subentry

1. PKCS#12 keymaterial is stored in the Directory as the value of the attribute ldapOwnKeymaterial (this was the recommended way to store the keymaterial since Dirx Directory 8.2A):
   a. Dump the value into a file by:

      ```
      dirxcp> <DAP bind as admin>
      dirxcp> show <DN of the ldapSSLConfiguration Subentry> -attr
      LSOKE_FILE=ldapkeys.p12
      ```

      This command creates a file 'ldapkeys.p12' with the pkcs#12 content.

   b. Convert the file to PEM using the openssl command line tool installed with DirX Directory V8.6:

      ```
      openssl pkcs12 -in ldapkeys.p12 -out ldapkeys.pem -passin
      pass:xxx -passout pass:xxx
      ```

      This command creates a new file with the PEM formatted keymaterial.

      The password xxx supplied after the keyword 'pass:' in the passin option must match the password that was used to protect the PKCS#12 container.

      The password xxx supplied after the keyword 'pass:' in the passout option may differ, but if the same value as for passin is used, there is no need to change the content of the password file reference by the attribute Pkcs12PasswordFile.

   c. Load the content of the PEM file to the new attribute ldapOwnKeymaterialPEM:

      ```
      dirxcp> <DAP bind as admin>
      dirxcp> modify <DN of the ldapSSLConfiguration Subentry> -add
      ```

```
LSOKP_FILE=ldapkeys.pem
```

d. Delete the old Attribute (not mandatory but recommended)

```
dirxcp> <DAP bind as admin>
dirxcp> modify <DN of the ldapSSLConfiguration Subentry> -rem
LSOKE
```

If old PKCS#12 was stored in file

2. PKCS#12 keymaterial is stored externally in a file (say ldapkeys.p12) and is referenced by the value of the attribute ldapOwnKeymaterialFile (This is how the keymaterial was provided in DirX Directory versions older than 8.2A):

a. convert the file referenced by the value of the attribute ldapOwnKeymaterialFile: given the value of this attribute is the pathname ldapkeys.p12.

```
openssl pkcs12 -in ldapkeys.p12  -out ldapkeys.pem -passin
pass:xxx -passout pass:xxx
```

This command creates a new file with the PEM formatted keymaterial.

The password supplied after the keyword 'pass:' in the passin option must match the password that was used to protect the PKCS#12 container.

The password supplied after the keyword 'pass:' in the passout option may differ, but if the same value as for passin is used, there is no need to change the content of the password file reference by the attribute Pkcs12PasswordFile.

b. Load the content of the PEM file to the attribute ldapOwnKeymaterialPEM:

```
dirxcp> <DAP bind as admin>
dirxcp> modify <DN of the ldapSSLConfiguration Subentry> -add
LSOKP_FILE=ldapkeys.pem
```

c. Delete the old Attribute (not mandatory but recommended)

```
dirxcp> <DAP bind as admin>
dirxcp> modify <DN of the ldapSSLConfiguration Subentry> -rem
LSOKM
```

Ldapservers Keymaterial stored as file

For compatibility reasons the DirX Directory 9.1 ldapserver continues to support storing the ldapservers own keymaterial as a file and refer to that file in the attribute OwnKeymaterialFile (LSOKM) of the ldapsslconfiguration subentry. The content of the file must be PEM formatted.

2.3.3. Migration of ldapserver's SSL configuration attributes

Beginning with DirX Directory 8.2B the attribute ldapSupportedEncryptionStrength of the ldapSslConfiguration subentry has changed to a multi valued attribute with values that can be Cipherstrings "EXPORT", "LOW", "MEDIUM", "HIGH", "RSA" and "ALL". These are mapped on a list of ciphersuites in openssl. Ciphersuite names as supported by the openssl library version linked by the ldapserver. Extensibility for future openssl extensions DXD since 8.2B supports an new ldap extended operation to retrieve the list of ciphersuite names: dirxextop -t ldap_ssl_ciphernames

Default is the value "HIGH"

Ciphers configured in the ldapSupportedEncryptionStrength attribute are applied only for the security protocol versions TLS1.0, TLS1.1 and TLS1.2 . The new attribute supportedEncryptionStrengthExt must be used to configure the ciphersuites to be used with TLS1.3.

Attribute ldapSupportedSecurityProtocols of the LSCFG subentry: In DirX 9.1 this is a multi valued attribute with values "TLSv1.0", "TLSv1.1", "TLSv1.2" or "TLSv1.3".

All protocol versions in the range between the lowest and the highest version are configured as acceptable TLS protocol versions.

2.3.4. Adjustment of user policies

Due to the fact that the settings and default values for size limits in paged search operations have been changed since DirX V8.3, it is probably required to adjust the user policies in the root DSE. The total maximum number of entries returned in a search result for a paged search operation must be specified explicitly now if it should differ from the default value of 16384. Also the maximum number of entries per page of a search result for a paged search operation can be specified explicitly now. See the DirX Attributes and Syntaxes manual 1.7.19 for more details.

Example:

For the administrator of O=My-Company the size limit for search result is administered to unlimited (value -1). In older DirX version this size limit applies for paged search results and none paged search results. The user policy setting might look like this :

```
USP={USN={/O=My-Company/CN=admin},SL=-1}
```

Starting with DirX V8.3 the size limit for paged search results can be specified explicitly. Since it is not possible to express an unlimited size limit for the new PRSL (paged result size limit) parameter, it is required to specify a high integer value as a replacement for an unlimited size limit. The user policy setting might look like this :

```
USP={USN={/O=My-Company/CN=admin},SL=-1,PRSL=10000000}
```

Consider also that it is possible to specify the maximum of entries per page in a paged search. The default value for the new parameter SPL (single page limit) is 2048. If this value doesn't meet your requirements then you need to specify it also. The user policy setting might look like this :

```
USP={USN={/O=My-Company/CN=admin},SL=-1,PRSL=10000000,SPL=4096}
```

Please note that the settings of SL have no effect for paged searches, as well as PRSL and SPL have no effect on non-paged searches.

2.3.5. Migration of IDM SSL configuration files

The tokens used in the idmssl.cfg configuration file have changed.Up to DirX 8.7 the security protocol version and ciphers were configured with the tokens idm_ssl_protocol and idm_ssl_ciphers, these must be replaced by new tokens.See the description of the new tokens

- idm_ssl_protocol_min
- idm_ssl_protocol_max
- idm_ssl_ciphers_list
- idm_ssl_ciphers_suites

in the Administration reference manual chapter "2.9 IDMS Configuration and Key Material Files" in section idmssl.cfg.

2.4. Upgrade installation in a shadowing environment

2.4.1. Upgrade from DirX V8.2B or newer

Upgrading from DirX V8.2B or newer to DirX 9.1 requires no further migration steps with respect to shadowing.

You are free to start the upgrade on your master or on any of your consumers as long as you do NOT use UNIQUE indexes.If UNIQUE indexes are used and the agreement to consumer has "REPLS=TRUE" meaning consumer can replace supplier, then you MUST start the upgrade on the master.

2.5. Uninstallation

This section describes the uninstallation on Windows only.For Linux uninstallation, see section 2.1.3.6, "Deinstalling the first and the second DirX installation".

On Windows, the prerequisite for successful uninstallation of DirX is the presence of 64-bit Java Virtual Machine version 11. When this JVM is present, it should be possible to start the uninstallation from both the Add and Remove system settings or from the command-line by executing "Uninstall DirX Directory.exe" from the <dirx-inst-path>\UninstallerData directory.

If problems occur during uninstallation, refer to the "Known Problems" section 7.1.15, "Uninstallation on Windows" for possible solutions.

During uninstallation all files and folders below <dirx-inst-path> which were created by the installer will be deleted.This includes all private files and folders in these installer created subfolders. Private files/folders added directly under <dirx-inst-path> will be preserved during uninstallation.

2.6. Code Signing and Verification

At this release, signature files named <original_name>.sig are supplied for all executables and dynamic libraries delivered with DirX. These signature files can be used to verify that the code has not been altered or corrupted since it was signed. To perform the verification, use the verify.sh (Linux) or verify.bat (Windows) script in the folder $DIRX_INST_PATH/signing and specify the full path of the public key for verification "dxd_sign_public.pem" on the command line. This public key is contained in the folder $DIRX_INST_PATH/signing and can also be downloaded from the support portal.For further details, see the section "Code Signing Files" in the chapter "DirX Directory Files" in the DirX Directory Administration Reference.

3. Administration

This section provides information about the state of DirX after its installation on the Windows and UNIX platforms and the prerequisite procedures, if any, that you must follow prior to setting up the directory as described in the Administration Guide.

3.1. Database Administration

DirX cannot run before you have performed the database configuration and initialization.See the Administration Guide chapter "Understanding DBAM and Storage Management" and the Disc Dimensioning Guide for details.

After initial installation, you can set up the directory as described in the Administration Guide.Sample scripts for the initial steps are provided in the directory <install path>\scripts.

DirX uses the Windows event service to report status (NOTICE) and warning (WARNING) messages.You can view these messages with the Windows Event Viewer, selecting Application Log.

3.2. Administration on Linux

This section provides information about DirX administration on UNIX platforms.

3.2.1. General Procedures

After successful installation, the DirX directory service and the LDAP server will not be running.

You installed DirX under a user id, you must log in as this user to administer DirX.The DirX installation has included a call of the shell script .dirxrc into your .profile script to set or extend the environment variables DIRX_INST_PATH, PATH and LD_LIBRARY_PATH.

Please ensure that the following environment variable is set as follows:

```
LANG=C
```

If it is not set automatically, you should add this line to your .dirxrc file.

The following environment variables must be set as shown and exported:

```
LD_LIBRARY_PATH=/<dirx-account-homedir>/lib:$LD_LIBRARY_PATH
PATH=$PATH:/<dirx-account-homedir>/bin
```

Now you can start DirX with the command:

```
dirxadm -c "start"
```

You can check whether DirX is running using the following command:

```
$ ps -ea | grep dirx
```

After you have started DirX, you can set up the directory as described in the DirX Administration Guide. Sample scripts for the initial steps are provided in the directory <install path>/scripts.

3.2.2. Additional Procedures on SuSE Linux ES

Prevent Changes of Device Permissions.

A reboot of SuSE Linux results in changes to the permissions of raw devices in /dev. These changes are performed by the system.

In order to prevent changes during reboot, you have to extend the file /etc/init.d/boot.local with the following command:

```
chmod g+w <list of all devices used by DBAM>
```

Here is an example for such an extension to the /etc/init.d/boot.local file: # /dev/hdb5 and /dev/hdb6 are used by the DirX Service

```
chmod g+w /dev/hdb5 /dev/hdb6
```

3.2.3. Filedescriptor limits on Linux

On Linux the file /etc/security/limits.conf defines the Hard and Softlimits for the max number of Filedescriptors per process, e.g.

```
*           soft    nofile          8192
*           hard    nofile          8192
# End of file
```

As DirX Directory 9.1 processes are supporting up to 8k sockets the limits can be set as shown above.

3.2.4. Additional Procedures on SuSE Linux ES 15 or Red Hat Linux ES 8 and above

In newer Linux platforms (Red Hat Linux ES 8 and above, SuSE Linux ES 15 and above), in kernel version >= 4.18 and glibc version >= 2.15 a new memory management method was introduced which is sometimes referred as 'using per thread memory arenas'. With this new method the memory consumption of the DirX processes can increase rapidly, as the OS keeps a once used but already freed memory occupied for the process for possible later reuse. Number of those arenas defaults to (8 * number of cores) on a 64-bit Linux system. If the memory increase of the DirX processes is not acceptable in your environment, and you want a much lower memory consumption footprint for those processes, you can define a lower limit (e.g.: 2) for the number of usable memory arenas. To do so, you should add the following environment variable to your .dirxrc file and also export it: MALLOC_ARENA_MAX=2; export MALLOC_ARENA_MAX.This change will be effective after the next restart of the DirX processes.

4. Compatibility

This chapter provides information about compatibility to previous DirX releases.

4.1. DBAM database compatibility

4.1.1. Interplatform compatibility of DBAM databases

DBAM databases are not portable between different platforms per design. Also, databases created with an older 32-bit version of DirX Directory are not compatible with any 64 bit version of DirX Directory.

4.1.2. DBAM database compatibility with previous DirX version

The DBAM database structure starting from DirX 8.2B is fully upward compatible with that of DirX 9.1. An upgrade installation is possible without performing any DBAM database conversion procedures.

Due to functional changes, schema extensions and changes in the internal structure of shadowing status and journal files in the DirX server, additional migration steps may be required.Please refer to chapter 2.3 "Upgrade installation and database migration" and chapter 2.4 "Upgrade installation in a shadowing environment" for necessary migration steps.

4.2. Other Compatibility Aspects

DirX Directory 9.1 compatibility with previous DirX releases is detailed in the matrix below.

DirX Version	X.500 Protocols DAP	DSP	DISP	Protocol rpc[1]	Shadowing Agreements	Configuration	Scripts
V8.9	yes	yes	yes	no	yes	yes	yes
V8.10	yes	yes	yes	no	yes	yes	yes
9.0	yes	yes	yes	yes	yes	yes	yes

(1) rpc is used for the communication between dirxadm or DirX Manager and the DirX Servers and for signalling purposes between ldapserver and DSA.

The user password attribute is per default single-valued. It is possible to extend the schema in order to make it multivalued.

PreferredDeliveryMethod Attribute in ldap protocol: According to RFC 4517 the output of PreferredDeliveryMethod attribute values as ldap strings contain the sequence " $ " to separate the elements of a delivery method attribute. Former versions used the '$' character only. As input DirX accept both forms.

The format of DSA audit and LDAP audit files has been changed. Evaluation of audit log files is only possible with the actual dirxauddecode or dirxaudstatistics tool.

In DirX Directory 9.0 automatic verification of binary backups was introduced.As it’s described in the new features list, in DirX Directory 9.0 and later versions, unverified backups, erroneous backups and backups verified by an older product version cannot be loaded.Before loading a binary backup, created by an older product version, it must be verified by the current version of dbamverify. As new checks are usually introduced in new product versions, a verification done with an older version is not considered fully verified.Due to this, if there is a backup created with an older product version, then it must be fully verified (including all components) with the new version, even if it was previously verified with the older version.

5. Restrictions

This section describes the restrictions of this release of DirX.

5.1. General Restriction

This restriction relates to the use of external backup tools: Do not use non-DirX backup tools to back up the DBAM database.Only dirxbackup performs the necessary synchronization with other DirX processes using the database.Also DirX protects its database against getting inconsistent by delay of modifications being performed while a backup is created.

Using non-DirX backup tools that access DBAM database files or devices can produce inconsistencies in the database and/or in the dirxbackup archive file.This has in particular been observed, when a Microsoft Volume Shadow Copy Service (VSS) based tool was used to create backups of the DirX DBAM Files. It has been also observed, that third party backup tools not using Microsoft Volume Shadow Copy Service (VSS) can produce inconsistencies in the database even if the DBAM database files have been explicitly excluded from the list of the to be backed up files. Also backing up the files residing within the DirX installation folder can result in problems, as the DirX processes require exclusive access to some of its file (e.g. journals).

Use non-DirX backup tools only on files outside the DirX installation and configure the backup tool such that the partition containing the DBAM files is excluded from the backup.

5.2. DAP

For DAP, the following restrictions apply:

add entry	targetSystem is not supported the maximum number of RDNs is 24
modify entry	the maximum number of RDNs is 24
list	pagedResults is not supported
read	modifyRightsRequest and attributeSizeLimit are not supported
search	matchedValuesOnly and attributeSizeLimit are not supported

5.3. DOP

This version does not support DOP.

5.4. DirX Server

5.4.1. DirX server limitations

The following limitations apply:

- The maximum PDU size for a LDAP protocol request is limited to 32MB. If the DirX LDAP server receives a larger PDU then the request is rejected with a ‘LDAP protocol error’ prior to decode the request.
- The maximum number of RDNs within a DN in an addEntry or modDN operation is limited to 24.
- Values of Operational Binding ID in shadow Agreements The range of Agreement IDs that a user can specify in an sob create command is limited to the range from 1 to 9999. If an agreementID exceeds this range an error is returned in the sob create operation. The DSA uses IDs greater 10000 for internal purposes, i.e. for the agreements concerning the CDT Entry.
- The number of recurring attribute values in one directory entry is limited. The limitation is due to the maximum number of DBAM follow blocks of 8192. The maximum number of attribute values evaluates to blocksize in General device (KB) * 8192 / (raw) size of the attributevalue (KB). This limitation can be overcome by applying the attribute outsourcing feature (see chapter "1.1.2 attr (dirxadm)" in the Admin Reference Manual.

Attribute values with Distinguished Name syntax are stored separately in extra blocks if the size of the values exceeds 16k. The maximum number of values is 128 million for each attribute type.

- The number of recurring attributes with DN Syntax (e.g. member of the group objectClass) that can be handled in one operation (transaction) is limited due to the size of the DBAM cache. See the following computation of the required DBAM cache space under worst case conditions. dirxdsa : DBAM cache requirements for large groups under worst case conditions

A. Creation of a group;Adding member values to a group:

The creation of a group requires for each member attribute value the creation of a pseudo object, an update of the real object block the member attribute value points to and an update of a bit string in the attribute index if the member attribute is indexed. Since the DSA performs the creation of a group in a single transaction, this may require a large amount of DBAM cache. This requirement calculates the following formula :

```
  DIRX_DSA_CACHE_SIZE =
  (
    (RealBlockSize+64+
     NoOfMemberValues/8192)          // Size for the group
itself in KB.
  + (NoOfMemberValues/8)             // Total size for pseudo
objects in KB.
                                     // 8 pseudo objects fit
into a 1k block
  + (NoOfMemberValues*1)             // Total Size for attribute
index bit strings
```

```
                                        // An average bit string
size of 1k is assumed
                                        // But can be up to 32K.
                                        // Set to 0 if attribute is
not indexed.
  + (NoOfMemberValues*RealBlockSize) // Total size for real
object blocks in KB
  )*2.2/1024                            // Result in MB
```

Since the DBAM cache requirement can be very large, it is recommended to create a group with a small amount of member values. Enlarge the group with additional modify operations.

Example :
Creation of a group with 100000 member values.
The real object block size is 4K and the attribute member is indexed :

```
  DIRX_DSA_CACHE_SIZE =
((4+64+100000/8192)+(100000/8)+(100000*1)+(100000*4))
                          *2.2/1024 = 1101MB
```

For the creation of the group a cache size of 1100MB required. The same amount is also required for adding 100000 values to an existing group.

B. Deletion of a group;Removing member values from a group:

The deletion of a group requires a large amount of DBAM cache if the member attribute is indexed. For each member attribute value an update of a bit string in the attribute index is required. For the deletion of a group or removing member values from a group the formula looks like this :

```
  DIRX_DSA_CACHE_SIZE =
  (
    (RealBlockSize+64+
     NoOfMemberValues/8192)          // Size for the group
itself in KB.
  + (NoOfMemberValues*1)             // Total Size for attribute
index bit strings
                                     // An average bit string
size of 1k is assumed
                                     // but can be up to 32K.
                                     // Set to 0 if attribute is
not indexed.
```

```
  )*2.2/1024                              // Result in MB
```

Example :
Deletion of a group with 100000 member values.
The attribute member is indexed :

```
  DIRX_DSA_CACHE_SIZE = ((4+64+100000/8192)+(100000*1))*2.2/1024
= 215MB
```

For the deletion of the group a cache size of 215MB required. The same amount of DBAM cache size is required for removing 100000 values from the group.

- Maximum Size of one single Memory Context A Ctx-Hard-Limit has been introduced (controlled via env DIRX_CTX_LIMIT) that sets the max upper limit for the entire sum of all CTX memory contexts within a process.

  A Ctx-Soft-Limit has been introduced (that is 40 MB below the Hard-Limit) and returns a NULL pointer to the caller when the Soft Limit is reached. Further allocations (up to the Hard-Limit) are serviced correctly. When the Hard-Limit is reached, a permanent error CTX_ULIMIT_EXCEEDED is returned.

  A Single-CTX-Limit is introduced that controls the max size of a single memory context (CTX consists of 0-N such single contexts). The single limit is set by default to Hard-Limit/2 on 64 Bit systems. If this limit is reached a CTX_LIMIT_EXCEEDED error is returned.

  By setting the DIRX_CTX_LIMIT environment variable the Hard-Limit can be controlled. By setting the DIRX_CTX_LIMIT_SINGLE env variable the Single Ctx Limit can be controlled. The Soft-Limit cannot be set (always Hardlimit-40MB).

- If both DIRX_CTX_LIMIT and DIRX_DSA_CACHE_SIZE are set by the user, the sum of the values cannot exceed 100% of the physical memory size. If none of the variables are set, the DSA calculates the CTX limit and the DBAM Cache size automatically based on the total physical memory. The automatic calculation allows only 80% of the physical memory to be used by the DSA and the LDAP server, 20% is left for the OS and for other applications. This 80% of memory is distributed between DSA(75%) and LDAP server(25%). If both DIRX_CTX_LIMIT and DIRX_DSA_CACHE_SIZE are set by the user, the sum of the values cannot exceed 100% of the physical memory size. If only one of the two variables is set, the other is calculated based on the prevously mentioned auto-calculated memory limit for the DSA. If any of the two variable (or both) are set, the DSA will check the values against the physical memory. If the values are not suitable, the DSA will not start and an error message will be added to the fatalDSA* logfile, containing the accepted value range based on the physical memory of that server.
- In the past an attribute with Directory String syntax could only store characters with 3 bytes or less, even if UTF-8 representation choice was configured. The DSA internally stores the normed charaters of a Directory String as 16 bit number. As an UTF-8 character consisting of more than 3 bytes cannot be represented using a 16 bit number,

the operation to add such data to the database was rejected. This behavior has been changed starting from version DirX Directory 9.1: 4 bytes UTF-8 characters (e.g.: emojis) are also accepted in attributes with Directory String syntax, but only with several restrictions:

  - Only the storage and retrieval of such characters is supported within the DB (dirxbackup, dirxload, ldif_dump, dirxmodify handles 4 bytes UTF-8 characters as they were entered into the DB)
  - dirxcp and dirxadm - because of the underlying Tcl library restrictions - cannot handle 4 bytes UTF-8 characters
  - Internal storage of normed character representation remains a 16 bit number to maintain DB compatibility. Because a 4 byte UTF-8 character cannot be represented using a 16 bit number, every such character will be normalized to the '.' character (this affects sorting, indexing, filter match).
  - As a consequence, a search result may be incorrect if 4 byte UTF-8 character is present in a search filter: value<emoji-1> would match to value<emoji-2>. Because of this behavior, search filters containing 4 byte UTF-8 characters will be rejected.
  - As another consequence, two values which only differ in 4 byte UTF-8 characters cannot be added to the same attribute.
  - You cannot have two entries under the same node, where the DNs differ only in 4 bytes UTF-8 characters
  - 4 byte UTF-8 characters can only be used in a shadowing configuration, when all the servers are already updated to a SW version supporting this. Otherwise the total update will fail, shadowing will not work.

- Maximum Number of filter items in a search operation A new limit has been introduced (controlled via environment variable DIRX_SEARCH_FILTER_ITEM_LIMIT) that sets the max upper limit for filter items contained in a search operation. If this environment variable is not set, the default limit value of 5000 filter items is used. If the filter item count exceeds the predefined limit, the DSA rejects the operation with 'unwilling to perform' error, and a diagnostic message about the limitation.

5.4.2. DirX server restrictions

The following restrictions apply:

- ModDN operation:
  - If a ModDN operation results in adding an entry to or removing an entry from a replicated area defined in a shadowing or ldif agreement, the DSA reports an "affects multiple DSAs" error.
  - ModDN operations are rejected during a total update shadowing operation is performed. There are 2 exceptions to this rule:
    1. ModDN operations are allowed during the particular total update performed as a result of a shadow administration command (total update of the replicated area /cn=cooperatingDSAs-Subentry).
    2. ModDN operation that do only change the last RDN (newSuperior is not set) are allowed during any total update.

- DIRX_SyncFile and shared filesystems
  The synchronization mechanism used by DirX to control access from the DSA process and/or other tools to the DBAM Database is a file named <installation path>/server/conf/.DIRX_SyncFile. This file must be located on a local file system. In case the installation path points to a shared file system perform the following procedure for the DBAM initialization:
  1. perform dbamboot: this tool will report an error, but will nevertheless create a correctly formatted .DIRX_SyncFile.
  2. copy this file to a local UFS filesystem
  3. create a symbolic link named <installation path>/server/conf/.DIRX_SyncFile to the copied file
- Max number of filedescriptors for DirX processes on Unix

  In DIRX the file descriptors 0-255 are reserved for internal usage and therefore all socket descriptors are shifted beyond 255. If this is not possible the following error output is displayed on stderr:

```
  ### IDM-Problem (error = 11523 - Cannot shift socket beyond 255)
  MoveDescriptorUp() failed in IDM_SetupActiveConnection(). Socket
Closed.
```

  This failure occurs, if you don't have enough descriptors available beyond 255. In order to prevent this failure, DirX binaries set the number of available filedescriptors to the value of 8192. In case the setrlimit() call fails the binary does not start and the following messages are printed to stderr:

```
   "Setting resource limit (NOFILE) failed!\n
    Softlimit: <old> -> <new>, Hardlimit: <old> -> <new>"
   "Increase the hardlimit to at least 8192 by adding
    set rlim_fd_max = 8192
    in /etc/system and reboot the system!\n"
```

  The actual limit can be checked with ulimit -n

  Starting with DirX Directory V8.4, on Linux the maximum number of filedescriptors for DirX processes is no longer restricted to 1024. The DirX sources are now compiled with increased FD_SETSIZE to 8192.

- The following max numbers of parallel incoming LDAP connections are supported by DirX:

On LINUX:	max 4000 LDAP Connections
On WINDOWS:	only limited by OS

The limitations are due to the fact that each LDAP connection may consume 2 descriptors (1 for the frontend LDAP connection and one for the backend DAP connection to the DSA).
A few descriptors are also consumed by the servers itself for internal communications. These figures describe the limits for a single ldapserver process.

- maximumvalue of a socket descriptor: DirX cannot handle socket descriptors that exceed 65535.
- On Linux platforms DBAM devices must be block devices. These devices can be configured using a logical volume manager, e.g. LVM under YAST on SUSE LINUX.
- Transaction log device and DBAM Cache Size: The DBAM Transaction log device contains transaction logs and is always a separate raw device. See the Disc Dimensioning Guide for details. The minimum size of the transaction device is 256 MB. For best performance the size should increase with DBAM cache size and checkpoint size. As a rule of thumb multiply DBAM cache size by 2 getting the optimum size. A transaction device size of 4GB performs well even with very large DBAM cache sizes of 4GB and a lot larger.
- Network connections:
  The server (and the client) can perform network connections either over IDM or over the SSL protected variant IDMS. DirX is prepared to work over IDM and IDMS.

that no interoperability tests with other vendor's X.500 products have been performed yet over the IDM/IDMS stack.

- Virtual list view (VLV) search operations are not supported, the associated ldapControl has been removed from the LDAP root subentry cn=ldaproot.
- Indexing (dirxadm db attrconfig operation)
  - optread option is not supported This option can be specified optionally when performing the db attrconfig operation. The value has no effect how the attribute is stored in the database.
- Shadowing and LDIF agreements
  - The consumer machine must be at least as powerful as the supplier machine.
  - Your system resources affect the maximum number of entries distributed over a total update by the DSA when performing DISP. Consider to configure total-update-by-media (CO=TRUE) for directories with a very large number (e.g. more than 1 million) of entries.
  - The number of shadowing and LDIF agreements is limited to 128.
  - Your system resources - for example main memory or disk space - limit the generation of LDIF files. To avoid problems when writing LDIF files enable automatic LDIF file splitting by setting the system environment variable DIRX_LDIF_SPLIT.
  - The server ignores values specified in the -validity option of the sob / lob create operations.
  - The sob / lob modify operations are not supported.
  - The MAXOSK component of the supplier policy (SUPP component of SOB-Policies

attribute syntax) is not supported.

- Specifying the value 0 or a value exceeding the available memory of your system for the MAXLOC component of the SOB-Policies attribute syntax is prohibited.
- When performing a total update by media that is using dirxbackup to restore the data saved on the supplier DSA to the consumer DSA the following applies:
- There is only one supplier DSA holding all master entries and all other DSAs participating in the network are consumer DSAs (contain only shadow entries).
  - The whole DIT should be shadowed on all consumer DSAs.
  - The supplier DSA and the consumer DSAs must run on the same platform.
  - The supplier and consumer DSA must have the same database configuration (DBAM profile).
  - the procedure described in chapter 10.3.2 "Performing a Total Update by Media" the Administration Guide must be followed.
- When terminating or deleting a shadowing agreement where the option CHANGEO=TRUE (changes only configured TRUE) is configured, shadowed entries remain untouched at the consumer DSA. That is if entries are deleted on the master DSA and afterwards the shadowing agreement is re-establishing respectively re-creating those changes are not propagated to the consumer DSA.
- If an administrator terminates a shadowing agreement where the total update has been performed by media and re-establishes the shadowing agreement a new total update is required.
- If you perform a dirxadm sob establish operation check the status of the corresponding agreement after about five minutes. Repeat the operation if necessary.
- In the event that the DSA has disabled an agreement the administrator must perform a dirxadm sob enable operation after having solved the problems occurred.
- If the number of updates exceeds your system resources, for example file size limitations (journal files) or available memory, and cannot be sent to the consumer the shadowing agreement is disabled. Then the administrator must perform a total update.
- The dirxadm sob switch operation can only be performed for all shadowing agreements and the whole DIT. In a scenario with asynchronous and synchronous shadow agreements only the latter Consumer DSA can replace the supplier.
- If the DSA crashed while performing update operations a consumer may be out of sync and a new total update must be performed after performing the emergency switch and restarting the former supplier. This does not apply to synchronous consumer DSAs.
- Secondary shadowing is not supported

- Password Policy
  - The compare operation is not Password-Policy aware.
  - In a Password policy aware system only single-valued user password attributes are allowed. No considerations are done for directories or systems that allow a user to

maintain multivalued password attributes.

- The operation to change a password that has been reset by an administrative command is a modEntry with a single change, a replacement of the userPassword attribute.
- The addEntry operation is not passwordPolicy aware, i.e. if the entry is created with an userPassword attribute, the password value is not checked against the password policy with respect to the password quality. The operational attribute pwdMustChange may be set to TRUE in order to enforce the change to a password policy conformant password value. = Enabling Global Password Policy State in a running system Switching to the Global Password Policy States requires to set the operational attributes to consistent values on all participating DSAs. Therefore, the following steps should be performed in the given order:
  a. disable the password policy features Account lock and Password Aging: modify all related password policy subentries by setting PPMAXA=0 and PPLOCK=FALSE
  b. set the environment DIRX_GLOBAL_PPO_STATES=1 on all participating DSAs.
  c. restart the DirX service on the Supplier and on all participating Consumer hosts.
  d. terminate and re-establsh the Shadow Agreements to all participating Consumer DSAs.
  e. enable the required Password Policy features. The steps a. and d. can be omitted, if neither Account Lock nor Password Aging were ever enabled in the directory service.
- If Global Password Policy States is enabled, the DSA Audit will contain records of modify operations that are not triggered by client application requests. These records represent the modifications issued internally by the DSA in order to change the password policy related state of an entry. The modifiers name for those operations is <DSA DN> suffixed with "--PPO"

- DSA Policies
  - If a shadow DSA is setup by media, it will that way be supplied with the DSAPolicy attribute of the root DSE. In case the Shadow DSA is setup by a total update by DISP protocol, the DSA Policy attribute must be administered explicitly at the Shadow DSA. Otherwise, future bind operations between master and shadow DSA will fail. This applies only to the case, where non-default DSA Policies are used.
- Server Side Sorting
  - The DirX DSA supports Server Side Sorting only for one attribute. This attribute must be indexed and must have an ordering matching rule. For non-reverse order sorting the INITIAL index, for reverse order sorting, the INITIAL and ANY indexes must also be configured. Attribute with string types are sorted implicitly lexicographically. Sorting in reverse order is supported. In dirxcp the server-side-sorting control is attached to a search operation by specifying the option -vsortkey <attrType> and -vtype vlv. Note that the vlv type must be specified for historical reasons and does not include the correspondent ldap control.
- Paged Search Operations and Chaining
  - Paged searching works in general only for not-distributed directories. If the base object of the search operation is the reference or is located below the reference,

then the paged search is executed by the remote DSA and the operation can perform. Otherwise, if the base object of the search operation is situated above a reference in the DIT, the DSA returns an "unwilling to perform" error.

- The matching rules for a given attribute must be consistent with respect to the case handling in the schema. That is, the matching rules for equality, substring and ordering must be either all ignorecase or all exactcase. The DirX DSA rejects schema entries with conflicting matching rule definitions, e.g. an attribute with DirStr Syntax cannot have an caseExactMatch equality matching rule and a caseIgnoreSubstringMatch substring matching rule. In this case an "unwilling to perform" error is returned and the Error Message is set to "Error in attributeTypes : Value=<AttributeType>; Reason=Combination of matching rules is not supported."
- Restart after restore DB
  The dirxdsa process is restarted after a database was restored by dirxbackup -R. The watchdog processes (dirxdsas / dirxsrv.exe) will restart the service and all components will have the knowledge of the restored DB.
- Sizes displayed by the tools dirxbackup and dbamdevinfo
  When performing the dirxbackup -T -i <archive_name> operation to display the profile info of a saved database, the tool reports device sizes for the various block types and the portion of the respective device that is used. The latter is referred to as "device range in use". The "device range in use" cannot be compared directly with device "space in use", that is part of the output of the dbamdevinfo command. The "device range in use"-size is computed based on biggest block number found in the respective device, whereas the "space in use"-size is the sum over all blocks used within a device.
- Ldapserver from a DIRX-CL package:
  Synchronization between a remote DSA and a ldapserver that is installed from a DIRX-CL package does not work correctly: If the DSA crashes, the respective DirX Service Process (dirxdsas or dirxsrv.exe) cannot notify the remote ldapserver process. Therefore, the ldap result cache may be used as long as there is no need to perform a rebind to the DSA. Advice: Do not use the ldap cache functionality on DIRX-CL package.
- Ldap result cache restrictions
  Search Results are not cached by the ldapserver if one of the following attributes occurs in the search filter or in the requested attributes list:

  pwdExpirationWarned

  pwdAccountLockedTime

  pwdFailureTime

  pwdGraceUseTime

  pwdFailureTime
- LDAP result cache size restriction
  The size of the LDAP result cache must not exceed 2 GB. The environment variable DIRX_WDOG_RESTART_LDAP_ON_DSA_RESTART=0 must not be set if LDAP cache is enabled.
- On 64-bit Plattforms the OSI Stack is not supported. This applies to the native 64-bit applications of DirX on Windows and Linux.
- Restriction that apply when using DNS Names in PSAP Addresses

Using DNS-Names in PSAP addresses (NAT support) is ONLY supported if the following environment is established:

a. all specified DNS-names must be administered in a way that the correct IP addresses are returned, i.e. both sides of a connection must be correctly administered in DNS.

b. Multiple IP addresses may be returned by a lookup of a configured dnsName from DNS. The DirX Processes use the first address returned by the name resolver for the specified IP version.

c. DirX must be installed and running on the IP address returned from DNS.

d. dnsNames are only supported for IDM communication stack, not for OSI

- Note that subschema subentries (Structure and Content Rules) are not needed.
- RACF external authentication
  RACF external authentication works only via IPv4.
- RPC protocol restricted to IPv4
  Internal RPC between DSA and LDAPServer process uses IPv4 stack only. If the IP stack for RPC connection is restricted to v6, the ldapserver won't start, the exit code in this case is 21.
- Progressive purge

  Some restrictions apply while progressive purge is running :

  - A normal purge (dirxadm command : db purge -pob) operation is rejected
  - A purge analyse (dirxadm command : db purge –analyse) operation is rejected
  - Post indexing (dirxadm command : db attrconfig ...) operation is rejected
  - ModDN operations are rejected if a new superior of the entry is specified.

  Starting a progressive purge is being rejected if one of the following operations are running :

  - A normal purge (dirxadm command : db purge -pob) operation
  - A purge analyse (dirxadm command : db purge –analyse) operation
  - Post indexing (dirxadm command : db attrconfig ...) operation

  See also the known-bugs entry concerning the progressive purge operation

- Dynamic groups
  dirxMemberUrl does not support host name, port number and attributes. Values of the attribute dirxMemberUrl must supply base-dn, scope and filter. In case the search base is the ROOT Dse then base-dn can be omitted. Extended filter match is not supported Phonetic matching is not supported Some attributes are not supported in filter items. These attributes are :

  - dirxMemberOf, dirxMember, numSubordinates, numAllSubordinates, structuralObjectClass, dseType, vendorName, vendorVersion, entryACI, prescriptiveACI, subtreeSpecification, subentryACI, all knowledge attribute,

password policy attributes and others.

Look into the schema file created by the DSA at startup time. There is table named "Allowed index and outsourcing configuration". Each attribute with a 'G' flag specified is not allowed for dynamic group filter expressions.

- LDAP Group Policies
  When using the ldapGroupPolicies configuration attribute to set rules for members of groups, the number of members is limited to 1 million per group. If a group contains more members the LDAP server will discard the rule. Escpecially when dynamic groups are involved, care must be taken to not exceed this limit by 'simple' filters that result in large member counts. The limit was introduced to avoid long startup times, as all members of LDAP group policies rules must be evaluated/read at startup time.Both, startup time and memory consumption will suffer, if the groups become large. Further, keep in mind that due to the fact that the members are read at startup time (or at time when the LUP is refreshed by an extnded operation) a change of the group members after startup or refresh time is NOT automatically recognized by the server. If frequent updates are required we recommend to trigger a dynamic LDAP User and LDAP Group Policies update by a cron job or extend client software (e.g. Workflows) to perform such an update after finishing their work.

5.5. LDAP Proxy

This paragraph refers to the DirX ldapserver running in LDAP proxy mode

- SSL/TLS Client Authentication
  If the LDAP server is configured to run in proxy mode, it will support SSL/TLS client authentication (CLA) only between client and proxy, but NOT between proxy and target LDAP server.This is due to the fact that CLA requires client key material to be present at time of connection establishment, which is not the case when the proxy will connect to the target LDAP server.Therefore, only SSL/TLS server based athentication is possible.
- Change of the Proxy Servers Configuration
  Usually the proxy reads its LDAP-configuration from the LOCAL DSA but not from any of the target servers DSAs. Thus, changes intended for the proxy must be done to this DSA and not to a backend DSA unless the proxy’s DSA is a full shadow of the backends.
- Connection limitation
  The proxy server will open one backend connection for every frontend connection, i.e no backend connection sharing is available, which must be considered rearding file descriptor usage. The proxy will be capable of holding up to 4000 parallel client connections on Linx. On Windows the number of available descriptors is not configurable and depends on OS and available physical memory.
- ldap extended operations
  By design, the proxy server will not forward Ldap Extended Operations with the only exception of RFC3062. This implies that Extended Operations used for monitoring will return the information from the proxy server itself and/or the LOCAL DSA but not from any of the backend servers.
- Proxy configuration file and UTF-8 characters
  The proxy server configuration consists of a JSON formatted file.As attribute values or

distinguished names have to be specified, a UTF8 supporting editor should be used to create that configuration file.Furthermore, the internal normalization of the configuration strings does not work for characters > 0x7f (multi-byte chars).Thus, if multi-byte UTF8 chars appear in the JSON file (e.g. small 0xC3 0xBC for the Latin1 'ü' character, they will not match to their capitals and must therefore be configured exactly as they will appear in the LDAP protocoi.E.g. let's assume a user containing a german Umlaut 'ü' (e.g 'Müller') shall have a explicit USER-Rule in JSON, then the user-name must be specified by the following 7 bytes M0xC30xBCller.This rule will then match whenever a LDAP user will bind as 'Müller', but it will NOT match if the user binds as 'MÜLLER' due to the fact that during normalization 'Müller' will be transformed to M0xC30xBCller and will match the rule while 'MÜLLER' will transform to M0xC30x9Cller and will not match. To overcome this flaw, it is possible to create another rule (copy) where the user-name appears with the capitalized chars.

And the implicitly performed case ignore match between strings from the incoming ldap protocol and the cofiguration values is not performed for multibyte characters.

5.6. dirxload

Dirxload option -x no longer supported. This option was intended to direct dirxload not to perform attribute indexing during the load operation.As it was up to the administrator to perform the post indexing after the load process, the attribute indexes became potentionally inconsistent with the loaded data.

In the default schema of DXD 8.7 the attribute type supportedSecurityProtocols is set to multi valued.Hence, if an LDIF file is loaded with a schema entry containing the old (single-valued) definition of that attribute an error occurs during the load, i.e. changing that attributeType definition of supportedSecurityProtocols to multi-valued as the input is required.

5.7. dirxadm

The maximum number of subentries displayed by dirxadm in a search result is limited to 128.

There is a bug concerning the deletion of multple attribute values, if the concerned attributes have a DistinguishedNameFirstComponent Equality Matching Rule.This is the case for values of the attributes UserPolicies (USP) and DSAPolicies (DSAP).As workaround you can perform the desired deletions in multiple operations, i.e. instead of

```
dse modify / -rem {USP={DN1};{DN2}}
```

use

```
dse modify / -rem {USP={DN1}} ; dse modify / -rem {USP={DN2}}
```

5.8. dirxcp

Certificate database files for dirxcp.

The Certificate database file is a NSS cert8.db (suitable for libldap60.so or nsldap32v60.dll).

5.9. DirX Manager

Enabling encryption with 256 bit keys for TLS/SSL

By default JAVA 8 only runs a weak-level cipher policy, ending up with using only export or 128Bit ciphers. To enable higher ciphers (like AES256) it is necessary to install a new policy package called

> Java Cryptography Extension (JCE) Unlimited Strength

This package actually replaces 2 policy files in the lib/security folder of the corresponding JRE installation.After doing so it is possible to run a 256Bit AES connection from DirXManager to the ldapserver.

Please refer to the file Documentation\DirX\DirXManager\Readme.html for further information on restrictions.

The port the snmp trap view window in DirX Manager listens on incoming SNMP V2 traps cannot be configured.DirX manager is always listening on port 162.

5.10. Support of IPv6 Addresses

General Notes on IPv4 / IPv6 communication restrictions:

a. only the IDM stack provides functionality for IPv6, the OSI stack does not.
b. For IPv6 traffic, it is highly recommended to run all DSAs on Dual-Stack machines, i.e. machines that provide the IPv4 and the IPv6 stack, because this provides the maximum flexibility to interconnect different types of DSAs.
c. The setting of DIRX_IP_STACK defines for which incoming traffic the DIRX servers will listen. The settings within the PSAP address defines which stack the DSA will use when he connects to other DSAs.
d. Currently all Redhat/Suse Linux are able to run as dual stack hosts (it might be necessary to activate the IPv6 stack explicitly, but it is basically there).Windows Server and new also provide dual IP stacks. WARNING: Don’t set the DIRX_IP_STACK variable to a value that your host system does not provide! (e.g. don’t set 'all' if you only have IPv4)
e. When IPv6 communication shall be used between two machines A and B, it must be ensured that the DNS resolver returns the IPv6 addresses of B when it is called from within A and vice versa for B (i.e. you must have a correctly configured DNS resolving that supports IPv6).
   If dual-stack communication shall be used DNS must return both IPv4 and IPv6

addresses of the dual-stack machines. You can check this by using the dirxhostinfo -I command.

6. Notes

This section provides miscellaneous information about DirX that is not provided in the DirX user documentation.

6.1. Example Scripts

This section provides information about installed example scripts.

6.1.1. Easy Use of Example Scripts

For the Entrust-Setup implementation some changes of the example scripts (Stand_alone and Entrust) were required.To avoid installing the same example files twice (for entrust and for Stand_alone) a file was created that includes all global variables, for example country, organization, contextPrefix, ...

All stand-alone and Entrust example scripts include the file
`<install path>/scripts/InitVar.tcl`.
This TCL-Script includes the file
`<install path>/scripts/Stand_alone/GlobalVar.tcl` or
`<install path>/scripts/LdapApplications/Entrust/Security-Manager/DefEntrustVar.tcl`
specifying all global variables for the setup.

If you want to use a private copy of the files and change some global variables you also may have to change the variables ScriptsDir, EntrustScripts, StandAloneScripts in the file
`<install path>/scripts/InitVar.tcl`.
Otherwise the default scripts under Stand_alone or LdapApplications/Entrust/Security-Manager are used.

6.1.2. Monitoring Scripts

This release provides scripts for monitoring a DSA.
The scripts are installed in
`<install path>/scripts/AuditingMonitoring`.

Please read the instructions in the file:
`<install path>/scripts/AuditingMonitoring/README`

6.1.3. LDIF Scripts

This release provides scripts to illustrate the use of LDIF agreements.
The scripts are installed in
`<install path>/scripts/LdifAgreements`.

Please read the instructions in the file:
`<install path>/scripts/LdifAgreements/README`

6.1.4. Shadowing Scripts

This release provides scripts to illustrate the use of shadow agreements. Note that shadowing can only work correctly if you have administered the environment variables `DIRX_DSA_NAME` and `DIRX_OWN_PSAP` correctly for all DSA participating in your network. That is NOT possible to work with the default DSA name and the local loop back address. The scripts are installed in
`<install path>/scripts/ShadowAgreements`.

Please read the instructions in the file:
`<install path>/scripts/ShadowAgreements/README_SHADOW`

6.1.5. LdapApplications Scripts

This release provides unintegrated scripts to illustrate the use of DirX in the context of PKI (public Key Infrastructure) and Access Management.

The scripts are installed in
`<install path>/scripts/LdapApplications`.

Please read the instructions in the files:
`<install path>/scripts/LdapApplications/*/README*` or
`<install path>/scripts/LdapApplications/*/*/README*`

1. The SSE scripts for TrustedCA support the PGP (Pretty Good Privacy) object class and attributes as well.

6.2. DirX Manager

The Java-based LDAP management client DirX Manager provides a configurable, platform-independent graphical administration interface for local and remote administration of DirX (and other LDAP compliant servers). DirX Manager provides its own installation procedure. Please refer to the Readme file Documentation\DirXDirectory\DirXManager\Readme.html for installation instructions and for information not provided in the DirX Manager online help and user documentation.

Installing and running the DirX Manager application requires a suitable Java Runtime Environment being installed on the machine as described in the Readme file. Java Runtime version 1.7.0_40 and later rejects server certificates during the SSL handshake, if the server certificate contains a public key with a length of less than 1024 bit. Therefore, generally, if a Java based LDAP client application (JRE >= 1.7.0_40) attempts to access a LDAP server using a public key with a length less than 1024, the LDAPS connection fails, if the ldapserver's keymaterial is less than 2014 bit. The SSL trace file shows a "certificate unknown" alert (0x2e) in this case.

The correctly working Java Runtime Environment should be checked by opening a command shell, and executing the 'java -version' command.

6.2.1. Activate console output redirection to file

In some cases additional debug traces are neccessary. For example for ssl problems you may want to set -Djavax.net.debug=all in the run.bat script for the DirX Manager.

If java instead of javaw is called inside the run.bat script all the logging goes to the console window.

For large output this is not very user friendly.

Now you can define an additional switch in the run.bat:

```
-Ddirxmanager.redirectconsole=true
```

Then all stdout and stderr output is redirected to a file.

The file(s) are named like dirxmgr_system.000.log

The number is incremented up to 999 and afterward a wraparound takes place. This means the files with lower numbers are overwitten. The files ar located in: `<instpath>/DirX Manager/logs`

The files are not deleted by DirX Manager. So the files have to be removed manually. Existing files are overwritten by the next run.

Possibly add some information from the following (source = stackoverflow):

Information on using the JVM flag

`-Djavax.net.debug=ssl`

all	turn on all debugging
ssl	turn on ssl debugging

The following can be used with ssl:

record	enable per-record tracing
handshake	print each handshake message
keygen	print key generation data
session	print session activity
defaultctx	print default SSL initialization
sslctx	print SSLContext tracing
sessioncache	print session cache tracing
keymanager	print key manager tracing
trustmanager	print trust manager tracing

pluggability	print pluggability tracing

handshake debugging can be widened with:

data	hex dump of each handshake message
verbose	verbose handshake message printing

record debugging can be widened with:

plaintext	hex dump of record plaintext
packet	print raw SSL/TLS packets

6.3. System Tuning

To accommodate heavy load conditions, that is, very large databases and/or a large number of clients, your server operating system should be tuned with respect to:

- unlimited file size for installation user id / root
- more than 512 sockets (NUMSOM)
- more than 128 network users (MAXUSERS)
- more than 500 semaphores (SEMMNU)
- number of file descriptors per process must be 8192 on LINUX.

6.4. Environment Variables

This section provides information how to set environment variables on different operating systems and on useful environment variables not documented in the user documentation.

6.4.1. DIRX_PROTECTED_ITEMS_LDAPNAMES

Optionally print out Protected-Items of ACI in ldap notation. Change Request.The dirxcp and dirxadm command line tools of DirX Directory 8.1A (and newer versions) support a new environment variable DIRX_PROTECTED_ITEMS_LDAPNAMES.If this variable is set with any value, the output of dirxcp and dirxadm in the pretty mode displays the attributes of the protected-Items within an ACI attribute (SACI, PACI or EACI) using the ldapnames instead of the abbreviations specified in the dirxabbr file. The default output without setting the environment is to use the abbreviation.

Example of default output:

```
...
User-Permissions
Protected-Items
Attribute-Type              : UP
: TN
```

```
All-Attribute-Values       : UP
: TN
Grants-And-Denials           : grantAdd+grantRemove
...
```

Example with set DIRX_PROTECTED_ITEMS_LDAPNAMES=1:

```
...
User-Permissions
Protected-Items
Attribute-Type             : userPassword
: telephoneNumber
All-Attribute-Values       : userPassword
: telephoneNumber
Grants-And-Denials           : grantAdd+grantRemove
...
```

The output of ACIs can only be customized for the pretty print mode, i.e when the option -p is specified in the search/show command.
As of DirX Directory 8.3 setting this environemnt result also in printing ldap-style ObjectClass Names contained in Specification Filters of (AcessControl)-Subentries.

6.4.2. DIRX_CONS_KEEP_REFERENCES

By setting the variable in the consumer dirxdsa environment to the value of TRUE (case-sensitive), the shadow consumer dsa is configured to ignore the following DSE-Types in incoming shadow update requests: SUBR, IMM_SUPR and XR. (SUPR is an attribute of the root dse and is therefore not a subject of shadowing). This applies to both the existence or the absence of such DSEs in the requests (would result in adding or deleting such entries).

6.4.3. DIRX_LDAP_MAX_PDUSIZE

By default, the maximum LDAP PDU size accepted by the ldapserver is 32 MB. The max size can be increased by setting the environment DIRX_LDAP_MAX_PDUSIZE=xx for the ldapserver process, where xx is the max size in MB.

6.4.4. DIRX_NO_SWITCH_RECOVERY

The switch recovery procedure is performed by each Supplier or Consumer DSA at startup time in order to confirm the current Supplier DSA has not changed since the DSA was online. In case network settings or firewall rules etc. prevent the DSP connection to be established or prevent an immediate connection failure, the DSA would hang in the DSP bind. Export this environment with the value 1 in order to skip the switch recovery bind at

DSA startup time. By setting this environment, the local DSA will assume that the Supplier DSA has not changed since it was online.

6.4.5. DIRX_MAP_CERT_ALTNAME_ATTR

DirX 8.10 introduces the possibility to configure the attribute used for mapping (see description of the attribute dapSaslAuthzIdMapping) by settting the environment DIRX_MAP_CERT_ALTNAME_ATTR to the OID of the mapping attribute. The attribute named by this OID must have an IA5Str Syntax. example:
DIRX_MAP_CERT_ALTNAME_ATTR=0.9.2342.19200300.100.1.3
During the sasl bind the user presents a certificate with an altName.email extension containing the value "abele@my-company.com". The DSA searches its whole DIT for an entry having this value in its rfc822Mailbox (with OID 0.9.2342.19200300.100.1.3) and assigns this entry as the bind requestor.

6.4.6. DIRX_SET_TLS_LEVEL_MIN and DIRX_SET_TLS_LEVEL_MAX

DirX 8.10 introduces the possibility to configure which TLS version are accepted by the dirx command line client dirxcp. The accepted range of the TLS version is by default set to TLS1.0 up to TLS1.3. This range can be restricted by means of the environment variables DIRX_SET_TLS_LEVEL_MIN and DIRX_SET_TLS_LEVEL_MAX. The valid values for these are "1.0", "1.1", "1.2" and "1.3".

6.4.7. DIRX_SYSSTART_TIMEOUT

Enhanced Service restart on Linux.The DirX service start mechanism on Linux implemented by the watchdog process dirxdsas has been enhanced.Dirxdsas will now use an incrementally enlarged waittime before killing and restarting the dirxdsa process.A DSA process may need some time to start up, if a database recoverey is neccessary.The watchdog will now take this into account by erforming at most 5 restart attemps in a loop, where the waittime is increasing from initially DIRX_SYSSTART_TIMEOUT seconds (default 30) up to (1+2+4+8+16) * DIRX_SYSSTART_TIMEOUT seconds.

The waittime applied to the dirxadm start command has been adopted.

The dirx watchdog restart loop in the dirx_start procedure of the $HOME/etc/dirx file is no longer needed and has been removed.

6.5. Access to DirX Using ADSI (LDAP provider)

Microsoft provides different ADSI versions for its operating systems. Access to DirX using ADSI works successfully from Windows systems with DirX Directory 9.1.

6.6. Deletion of profiles with dbamconfig

Deletion of the active DBAM profile is rejected by dbamconfig. dbamconfig -d -P <oldProfileName> is returning error, if the .DirXSync file contains the same profile name -P <oldProfileName>.This behavior was introduced to avoid deletion of the active profile accidentally, since the DBAM database couldn’t be accessed anymore after the deletion.

If it is required to change the device specification of the active profile perform the following steps :

- Make a backup of the database with dirxbackup if the database is to be reused.
- Create a new profile first with
  dbamconfig -c -P <newProfileName> <device-spec>
- Initialize the devices and the .DirxSync file with dbamboot -P <newProfileName>
  Now <newProfileName> is the new active profile.
- Delete the old profile with dbamconfig -d -P <oldProfileName>
- Optionally restore the saved database with dirxbackup.

6.7. watchdog start without (x)inetd on LINUX

The watch process dirxdsas on Linux may be started from a shell. After completing the installation, you can start the dirxdsas using the DirX installation account.
Syntax: dirxdsas -m -d <dirx-inst-path>

6.8. DirX crash handler

The settings for the DirX crash handler may have been changed.Please read the notes on this topic which you can find in the ReadMe.txt file (on Linux) or in the ReadMe_WIN.txt file (on Windows) installed in the crash directory.

6.9. DirX DSA cache size

DBAM cache is recommended to be set to the total size of the currently used database blocks or larger.Unless the complete database fits into the DBAM cache, some operations may have to read blocks from the disk which results in longer request durations.For example, if a search needs to read in 10000 1k blocks and the disk can read at that time with 10MB/s, then only disk reading will take 1s.

You can get total size of the used blocks by either creating a non-compressed backup and checking the size of the file or from the dbamdevinfo output.To calculate the total size you have to add up the used space in all logical device in the dbamdevinfo output.For example, 3.6 GB of an 8 GB REAL device, 9.2 GB of a 16 GB AVIDX device and 4.4 GB of an 8 GB logical device is used, then the currently used blocks in your database takes 17.2 GB.Now you should consider if your database will grow because more entries will be added or new indexes will be created in the near future.Here in this example a growth factor of 1.3 is considered, because it is assumed that 30% more entries will be added and no new indexes will be created.So the new size is 22.36 GB.

The value of 22.36 GB specifies the amount of memory which is required for data buffers to load the content from disk into the DBAM cache.But the DBAM cache also needs memory for the page headers, for buffer management and a reserve of data buffers for the page version management.So it is required to multiply 22.36 GB with a factor of 1.27. That is 28.4 GB.Since the unit for the environment variable is megabyte you need to define DIRX_DSA_CACHE_SIZE=28400 to specify the size of the DSA cache.

6.10. dirxschema tool

The dirxschema tool can be used to extract the currently used schema from a running DirX Directory environment as an ldif file.

USAGE:

dirxschema
[-h <host>] (Host name or IP address of LDAP server.Default: localhost)
[-p <port number] (LDAP server's port number.Default: 389)
[-s <schema name] (LDAP schema name.Default: cn=schema)

Example :

```
dirxschema -host <SUPPLIERDSA> -p 389 >supplier-schema.ldif
```

7. Known Problems

This section describes known problems in this release of DirX.

7.1. DirX Server

This section describes known problems relating to the DirX server.

7.1.1. Logging

You should avoid high trace levels like 5, 6, 7, 8, and 9 because the server might exit when running under heavy load.You should use these trace levels only temporarily.

7.1.2. Context Prefix

Creating a context prefix below an already existing context prefix in the same DSA will result in errornous behaviour. It is strongly recommended to avoid this.

7.1.3. Aliases and the number of Subordinates

When creating aliases both, the alias object and the object referred to by the aliasedObjectName attribute of the alias result in an increment of the corresponding number of subordinates. This affects the attributes numberOfSubordinates and numberOfAllSubordinates as well as the output of the dirxadm command "db check -bs subordinate" Note that this is independent of the fact whether the aliasedObject exists or not at the time the aliasObject is created.

7.1.4. Switching in a Multi Consumer Environment

In a multi consumer environment one master DSA is replicating its directory data to two or more Consumer DSAs.

When performing a dirxadm sob switch operation one of the consumer DSAs becomes the new master DSA. The resulting protocol exchanges (DISP) involve the distribution of the modified cooperating DSAs Subentry (/CN=Cooperating-DSAs-Subentry) to all consumer DSAs and publishes the knowledge of the new mastership to all participating DSAs.

In the event of a communication error distributing the update of the cooperating DSA table may be incomplete: The concerned consumer DSA gets no information about the identity of the new master. Therefore, this DSA does not accept changes from that new master DSA. On the other hand the former master DSA may have already switched to a consumer role. The DSAs run out of sync.

The old master DSA logs such a situation in the following exception message:

```
"Consumer DSA did not participate in switch!
 Agreement ID: <agreementNumber>
```

```
      Total Update required for this DSA"
```

This situation can be resolved by a restart of the DSA, that has missed the switch protocol exchanges.

The switch operation is rejected by the old master, if there are outstanding updates for a consumer DSA and the synchronization of these updates fails due to connection problems to that DSA.
The switch operation might be rejected by the master if there is a SOB agreement where total-update-by-media (CHANGEO=TRUE) is configured, the shadow was just loaded from a backup, and that shadow has not seen a single update at the time of switch.

7.1.5. Shadow/LDIF Agreements and DSA Names and PSAP addresses

This section provides information about known problems with the handling of DSA names.

7.1.5.1. Sob switch and Consumer DSA is Permanently Unavailable

If a DSA administered as a shadow consumer is permanently unavailable the respective shadow agreement is automatically disabled after the occurrence of a couple of connection errors. However, there is a special agreement with the same consumer DSA name and the replicated area set to /CN=Cooperating-DSAs-Subentry. Due to the existence of this agreement, the DSA rejects a dirxadm sob switch operation. In this case, the special agreement must be deleted manually using the dirxadm sob delete command.

Note that this situation may also occur if an incorrect (consumer) DSA name was specified while administrating the shadowing or LDIF agreement.

7.1.5.2. Backup generated by another DSA

After setting up a DSA from a backup that has been generated by another DSA and that contains shadow or LDIF agreements the DSA names occuring in these agreements do not match the local DSA's name.

The same happens if an ldif content file from another DSA is loaded with dirxload without specifying the option -s.

In these cases, the DSA does not start after restoring such a database and there is the following record in the exception file:

```
"Own DSA Name:
     <DSA-DN1>
 Not found in Cooperating DSA Table:
     <Role>: <DSA-DN2>"
```

In order to manually delete the shadow agreement configuration perform the following

steps:

1. Start a shell / DOS-Shell
2. Specify the value <DSA-DN2> for the system environment variable `DIRX_DSA_NAME in that shell.`
3. Start the DSA the administrative mode. From the shell type `dirxdsa -a`
   As a result the DSA process (dirxdsa) starts but does not listen for protocol messages that is it can only be administered via the dirxadm tool.
4. Use dirxadm to
   - modify the cooperating DSAs Subentry with the command: `dirxadm> remove_knowledges`
   - stop the service
5. Terminate the shell
6. Start the DirX Service.

7.1.5.3. PSAP Match function

When PSAP Addresses are compared, all Selectors (TSEL, SSEL and PSEL) are taken into account. On the other hand, for the communication over the IDM stack the Selectors are ignored. However, as the match function includes the selectors this may result in failures reported. Therefore, the correct administration of PSAPs including the consistent naming of the selector fields is required, even if all communication is performed over IDM.

Particulary, when shadow- or ldif agreements are created, the DSA checks its Role by comparing the PSAP of the master of the cooperatingDSAsSubentry with the Value of its environment DIRX_OWN_PSAP.

7.1.6. Peculiarities with Alias Searching

a. derefAlias=Never flag and sorting control

   Search requests with the serviceControl derefAlias set to "Never" cannot be sorted. Many client Applications send this derefAlias value, as Never is defined in ldap to be 0 (default). Sometimes it may be impossible to change the client applications in order to send the suitable derefAlias flag value. In these cases, as a workaround the respective value of the LdapSearchServiceControl (LSSC) Attribute of the LdapConfiguration Subentry may be set to 'x'. As a consequence the respective ldapserver overrules the actual value sent in the ldap search request and the search is performed as if the request included the flag derefAlias=Always. If there is a need to perform both, searches for Aliases (derefAlias=Never) AND searches for AliasedObjects (derefAlias=Always) multiple ldapservers using different configurations may be used.

b. derefAlias=Never flag in the absence of Aliases

   If databases do not contain any ALIAS object, the setting of the derefAlias is ignored. As a consequence Sorting is possible even if derefAlias=Never is set.

c. derefAlias=Never flag and search performance

   The database layout and the DirX DSAs search procedure is performance optimized for the case, that a search operation requests the dereferencing of Aliases, resulting in the return of the "aliasedObjects", i.e. a search with the derefAlias=Never flag is slower than the one with the derefAlias=Always flag.

7.1.7. RACF External Authentication

Documentation of RACF_LDAP_TIMEOUT: The configured RACF_LDAP_TIMEOUT is a connection timeout. It is applied to the bind operation sent to the remote RACF ldap server. In case of dynamic mapping this timeout has no effect on the remote search operation.

7.1.8. Status of Agreements with CHANGEO=TRUE

If a Shadowing Agreements is configured with CHANGEO=TRUE - i.e. the consumer is to be set up by media - the status of the agreement is automatically set to "disabled". This applies as well for an agreement that is created as COOPERATIVE and for an agreement that is set to COOPERATIVE with the "sob establish" operation.

Such agreements have to be enabled explicitly after the consumer has been loaded with a backup media.

7.1.9. Re-Establishing of Shadow Agreements

If the attempt to reestablish a shadow agreement is rejected with the error "unsupported strategy", the administrator should check the status of a special Agreement to the same consumer DSA. This special agreement has a agreement ID > 10000 and the replicated area is set to /cn=cooperating-dsas-subentry. If the state of this special agreement is "DISABLED", it must be enabled (use the sob -enable -agree 1000x dirxadm operation) prior to the establishment of the primary agreement.

7.1.10. QUE3 Search Engine

The QUE3 Search Engine is disabled by default, as it is not stable.

7.1.11. Audit files

The DSA does not start if there are 'unclosed' audit files (audit.log) from earlier versions of DirX. You must rename or delete these files.

7.1.12. Remaining CP DSEType for the Root DSE

If a Directory server was configured with synchronous shadowing agreements, the DSE-Type Contextprefix (CP) will remain at the root DSE even after all synchronous shadowing agreements were deleted. Note: the dirxadm procedure "remove_knowledges" removes all information related to shadowing agreements turns a directory database into a standalone directory. This procedure performs cleans also the root DSE.

7.1.13. Scheduled Shadow Agreements and sob sync/switch

When a scheduled agreement is created and established a BeginTime (BT) may be specified. In this case the agreement starts at BT with the transmission of a total update. From that time on, updates are collected in the Supplier's journal and replicated to the consumer according to the schedule parameters UpdateInterval (UI) and WindowSize (WS).

If a "sob sync" is executed or if the synchronization of the journal is triggered implicitely by executing a sob switch operation, a Total Update is sent immediately in the context of the sync. However, the replication of updates occurs not until BT has been reached. This may lead to inconsistent data on the scheduled consumer DSA. Therefore, it is important to avoid sob sync operations in case there exist scheduled agreements in the pre-active phase(the time between establish and BT). Likewise, creation of scheduled agreements with a future BT should be done AFTER the switch operation, if such a switch is planned.

7.1.14. Uninstallation on Windows

The following error message can appear during the uninstallation on Windows:

"Could not find a valid Java virtual machine to load."

The reason for this error message can be that Java Virtual Machine (JVM) is not installed or the uninstaller couldn't find it or the JVM's version is incorrect. See section 2.6, "Uninstallation" for the correct version of JVM and the instructions for the uninstallation. If this error message still appears, try the following procedures:

1. Open the file "<dirx-inst-path>\UninstallerData\Uninstall DirX Directory.lax" and set the lax.nl.current.vm property to:
   `lax.nl.current.vm=<path-to-java.exe>`
   <path-to-java.exe> specifies the path to the java.exe program which can be found in the JVM installation directory. For example:
   `lax.nl.current.vm=C:\\Program Files\\Java\\jre\\bin\\java.exe`

   

   Double backslashes must be used instead of single backslashes in the path. After making this change, it should be possible to start the uninstallation from both the Add and Remove system settings or from the command-line by executing "Uninstall DirX Directory.exe" from the <dirx-inst-path>\UninstallerData directory.

2. Open the command-line and go to the <dirx-inst-path>\UninstallerData directory. Run the following command:
   `"Uninstall DirX Directory.exe" LAX_VM <path-to-java.exe>`
   <path-to-java.exe> specifies the path to the java.exe program which can be found in the JVM installation directory. For example:
   `"Uninstall DirX Directory.exe" LAX_VM "C:\Program Files\Java\jre\bin\java.exe"`

7.1.15. Access Control Modifications

There is a known problem in the handling of modifications of access control information (ACI) items. Due to this problem, after the necessary modifications of ACI items (entryACI, prescriptiveACI or subentryACI) are finished, DirX must be restarted.

7.1.16. Modification of unreferenced attribute values with DN syntax

Due to the internal structure of the database, unreferenced attribute values with DN syntax cannot be modified if according to the matching rules the old and new values are equivalent. Note that this problem is relevant just for attributes which are not referenced (contain DN of non-existing entry).

For example, the following modifications will not be executed:

- `cn=dummy  name,o=my-company → cn=dummy name,o=my-company`
- `cn=dummy name,o=my-company → cn=DUMMY name,o=my-company`
- `cn=dummy name,o=my-company → cn=  dummy name,o=my-company`

In case of similar modifications, please follow this sequence

- remove the value (for example `cn=dummy  name,o=my-company`)
- execute db purge command in dirxadm
- add the new value (`cn=DUMMY name,o=my-company`)

Please note that db purge requires exclusive rights to the database so other modify operations cannot be done while purge is running.

7.2. LDAP Server

There are the following known problems relating to the LDAP server:

- The LDAP server does not start if the LDAP cache is enabled and an RPC port cannot be established.
- The LDAP Server rejects schema modifications for objectClasses if SUP objectClasses are specified that are not defined earlier. For example, if an object class or a superior object class is defined and modified in a single operation the definition of the object class must appear before the modify operation.
- The dirxadm ldap mib -current curr_lbinds command will return incorrect values if suspicious bind PDUs are rejected or SSL binds fails.
- The LDAP server does not start if startTLS is configured TRUE but the OpenSSL library is not initialized. To initialize the OpenSSL library, the PSAP Self-Address (dirxldap.cfg) must contain a SSLPORT>0 definition. This does not mean that a SSL port is established (can still be 0), but initializes the OpenSSL library for usage during startTLS processing. Please note that a startTLS is performed on 'plain' connections only and does not

require an established SSL port !

- The LDAP server behaves unpredictable if other applications
  open the audit file/files used by the current LDAP server process. This does not apply to dirxauddecode, i.e. it is possible to evaluate the audit trail of a running LDAP server process with dirxauddecode. Perform a dirxadm audit -move operation, if the binary audit file should be examined with a third party tool. Then the LDAP server operates on a new file while the tool operates on the file detached from the LDAP server process. Do not use third party backup tools on audit file that are still in use by the server (e.g. on WINDOWS some backup tools put an exclusive lock on backup files. This will cause problems when the server tries to write or wrap around those files).
- Audit modus -move" vs file system space
  write (xdr_write) may result in an error if the disk space is running low. This depends on a quota, i.e. even if tools like df show a certain percentage of available space the write call may fail. Administrators that use the -move strategy have to care for enough available disk space. This applies as well to the DSA audit.
- Audit Timers
  If DIR.X runs on a multi CPU or multi core CPU platform and more than 1 CPU is online for the DIR.X processes, the High-Resolution Timestamps (resolution < 1 usec) within the Audit (DSA+LDAP) might show incorrect values due to the fact that the Hires Times are taken directly from the CPU which is a separate counter for each CPU. These counters are not synchronized by the system by any means and can lead to situations where subsequent Timer calls are resolved from different CPUs (threads-scheduling) which in turn leads to incorrect duration values. (On single CPU machines or if only 1 CPU is online this problem does not occur). There is currently NO way to synchronize these Hires-Clocks by software and there is currently also no way to ensure that two subsequent calls for the Hires Timer are taken from the same CPU. To overcome this problem, it is recommended to set the environment DIRX_USE_WEAK_TIMER=1. This will switch the timers to a low-resolution clock (1-20 ms resolution) that ensures correct time values at the disadvantage of a lower resolution. This applies as well to the DSA audit. Per default the weak timer is used on Multi CPU machines, the high-resolution timers otherwise.
- UserDump/Process Dump on Win64 Platform
  On Windows Platforms the Userdump tool may be used to generate a user dump of a process that shuts down with an exception or that stops responding (hangs). As a better alternative the 'procdump' tool from Microsofts Sysinternals-SuiteS. 'procdump' is currently the preferred tool.
- LDAP server and retrieval of ldap assoc MIBs
  The retrieval of the LDAP assoc MIB is handled by the ldap listener thread. During the processing of the MIB, the listener thread is unable to accept new connections for a short period of time.Hence it is not recommended to retrieve the assoc MIB with a high frequency (more than once every minute). Also, simultaneous retrievals of the assoc MIB may result in crashing the LDAP server process.Similar problems may occur when parallel queries for the LDAP current MIBs are requested.

7.3. dirxdumplog

On Linux platforms, the logfiles processed by dirxdumplog may contain logentries where

the symbolic name of the return code of a system call is wrong.This is due to diverging system error value definitions in sys/errno.h.For example a bind systen call may be reported to return EPROTOTYPE instead of EADDRINUSE.

7.4. dirxmodify

Problems can occur if large schema modifications are contained within LDIF-change files.Use LDIF-content files or reduce the number of schema modifications per LDIF record.

The behavior of dirxmodify is undefined if LDIF files are processed in which records appear where the 'dn' is not the first line of the record.

7.5. dirxcp

7.5.1. Handling of Blanks in Filter Expressions

Applies to DAP binds only.If the filterexpression contains the OCL attribute trailing blanks lead to not recognizing the abbreviation:

```
search / -sub -filter {ocl=ORP && sn=foo}
Error: Abbreviation unknown - "ocl=ORP && sn=foo" : Error position:
"ORP && sn=foo".
```

If the filteritems are specified without blanks or with in reverse order result in a successful parsing of the input.

```
search / -sub -filter {ocl=ORP&&sn=foo}"
```

or

```
search / -sub -filter {sn=foo && ocl=orp}"
```

are both working successful.

7.6. openssl command line tool

Running the public domain command 'openssl' from the DirX installation shows the warning

```
        WARNING: can't open config file: /usr/local/ssl/openssl.cnf
```

The warning can be suppressed by setting the environment variable OPENSSL_CONF to a

file of zero size.

on Linux:

```
OPENSSL_CONF=/dev/null; export OPENSSL_CONF
```

on Windows:

```
set OPENSSL_CONF=NUL
```

7.7. dirxsupervisor

An update installation performed on Windows will overwrite the TCL script file setup_common_data.tcl.In this file users will have changed variable values according to their needs.Therefore - in case the dirxsupervisor is used - this file has to be saved before and restored after an update installation manually. On UNIX the update installation installs the file setup_cccommon_data.tcl from the installation set with the suffix .new.

7.8. Documentation

This section describes known problems relating to user documentation.

7.8.1. Administration Guide

The user interface of DirX Manager may have changed and may look different than what is described in the manual.

7.8.2. Disc Dimensioning Guide

If you cannot access the dimensioning spread sheet file "dimensioningSpreadsheet.xlsx" with the Acrobat Reader file attachment tool use the file from the product DVD.The file is located on the DVD in the directory Documentation/DirX Directory.

8. History of Previous Releases

8.1. DirX Directory 9.0

8.1.1. New Features

New functionality:

- LDAPServer now supports the 'SecurityLevel' feature of OpenSSL. (see: https://www.openssl.org/docs, search for set_security_level for details) Because of this the LdapSSLConfiguration subentry was extended with 2 new attributes: ldapSecLevelA and ldapSecLevelB.For details about these new attributes please refer to the Syntaxes and Attributes Manual, chapter 3.1.15 Attributes for LDAP Server SSL Configuration.
- Default schema has been extended with 3 new attribute types: dmdName, pseudonym and organizationIdentifier. For details about these new attributes please refer to the Syntaxes and Attributes Manual, chapter 3.1.2 X.500 User Application Attributes, Names in General.
- dirxbackup now can generate a full LDIF dump or an LDIF dump of a specified subtree from a binary backup. For details, please check the documentation of dirxbackup's -L command line argument in the Administration reference.
- ldif_dump and create_total_ldif now can dump a subtree as well. Both commands were extended with an optional subtree parameter to be able to define the starting point of the LDIF dump in the tree.
- dirxbackup's saving functionality was extended to automatically verify the generated binary backups. To preserve the previous behavior and only create the backup without automatic verification, the -n switch can be used.
- Binary backup headers were extended to contain verification information. From this version on, dbamverify and dirxbackup will write the result and some metadata of the executed verification to the backup file's header. This verification information is checked by dirxbackup's restore operation. If a backup was not completely verified or contains errors, then loading will be rejected.
- A new server, dirxhttp was introduced to the DirX Directory service. It makes possible to access the DIT using the HTTP protocol with a custom JSON schema. An interactive API documentation can be found at https://<server_ip>:8443/dxd/ldap/v1/doc
- New paging memory optimization was introduced. This optimized paging memory handling, aims to reduce the size of the paging memory context during paged search operations compared to the previous version especially in case of filters with a huge number of elements. The optimization can be disabled by setting the DIRX_PAGING_MEMORY_OPTIMIZATION environment variable to zero.

Diagnostics and logging: Several enhancements have been implemented to yield more diagnostic information

- SchemaChangelog: Provides information of all schema changes ever made since the DSA was first started with the currently running DB. The log information is stored in the hidden logfile <DIRX_INST_PATH>/server/log/.schema_changes.txt The logfile will only

be reset in case of a successful dbamboot or dirxbackup -R.

- dirxadm RPC operations targeting the DSA are now logged in the DSA audit files. Because of this new fature the RPC interface between dirxadm and DSA has been extended, so it is not possible to use an older version of the dirxadm tool together with DirX Directory 9.0.
- new log collector scripts available for both Windows/Linux versions to ease symptom collection in case of an incident. The scripts (dxd_diag.bat/dxd_diag.sh) can be found under tools\dxd_diag folder. See readme file in that folder for more details.

8.1.2. Discontinued Features

The LDAP Mib interface in dirxadm is deprecated.It will not be supported in future versions of DirX Directory.Use of the LDAP extended operations is recommended starting from version DirX Directory 8.4.

8.1.3. Changes to the User Interface or Configuration

No changes since DirX Directory 8.10. No new SNMP traps since DirX Directory 8.10.

8.2. DirX Directory 8.10

8.2.1. New Features :

New functionality:

- a new process DirX Progsvr for secure execution of PROG policy commands was created.A PROG policy is an external command performed after creation of an LDIF file, it is defined by an LDIF policy.Moving execution of PROG policies to a separate process increases the stability of the whole system. Progsvr contains a pool of worker threads for the execution of PROG policies.It must run on the same machine as the DSA process.For more information about Progsvr please refer to the "Introduction" handbook, section "1.4 The DirX Directory Progsvr".The list of environment variables used by Progsvr can be found in the "Administration Reference" manual, section "4.2 DirX Directory Environment Variables".New dirxadm commands which are used to control the Progsvr are described in the "Administration Reference" manual, section "1.1.9 progsvr (dirxadm)".Finally, files and folders used by Progsvr are listed in the "Administration Reference" manual, section "6 File Locations".
- dirxcp via LDAP supports a new command option -control to specify LDAP controls. The syntax is: -control controloid[,criticality[,value]] It allows to specify any LDAP control attached to all operations and an arbitrary control value. The control value can be even read from file, e.g. -control CONDOP,1,<C:\\path\\to\\file\\file.name> For more information refer to the "Administration Reference" manual, section "1.2.4 obj (dirxcp)" where the new option was added to several dirxcp commands. A detailed description of the -control option is then given in the subsection "-control Option".
- support tunneling dirxadm RPC operations over LDAP extended operation. New LDAP extended operation dsa_dirxadm_cmd with a specific OID was implemented that indicates to the receiving DSA that an LDAP client wants to execute a dirxadm

command that is given in the payload of the extended operation. For more information see the "LDAP Extended Operations" manual, section "1.3.4 dsa_dirxadm_cmd".

- paged search requests on synchronous consumers was made more reliable by storing initiator DSA's role in the query reference and executing subsequent next page requests on the initiator DSA.

Security:

- Per default, LDAP server accepts TLS protocol versions 1.2 and 1.3 only
- SELinux in unconfined mode: A new script was implemented which configures DirX Directory to run in unconfined mode and the Linux installer was extended, see the section "1.2.7 Software Requirements" for more information
- OpenSSL update to the 1.1.1m version

Support of additional proprietary LDAP Request Control:

- LDAP Matched Value Only Filter Control: A search operation returns only those requested attribute values that match to the filter in the control's value. For example, it can be used to retrieve one particular certificate. The OID of the control is 1.3.12.2.1107.1.3.2.12.13. For more information refer to the "DirX Directory Syntaxes and Attributes, Edition March 2022", the section "2.4.3 LDAP Matched Value Only Filter Control".

Performance: Multithreaded CheckPointing

- during a checkpoint operation the commited transactions are written to the data devices (reserved areas on physical disc)
- if more than one data device is configured, the checkpoint write operation is now multi-threaded per default (it was single-threaded in the past)
- default number of additional worker threads is 2

Robustness: enhancement of the Linux watchdog process dirxdsas

Validation of data:

- attribute and object class names can be checked for valid characters (disabled by default), see "2.1.2.3 Enable schema LDAP name checking according to standard"
- the dirxload utility rejects an LDIF file in case it finds a schema element with an empty description.

Diagnostics and logging: Several enhancements have been implemented to yield more diagnostic information

- enhanced ACI logging: provides an easier way to investigate complex access control decisions. Configured with new LDAP extended operations dsa_ac_log_on and dsa_ac_log_off. For more information see the "LDAP Extended Operations" manual, sections "1.3.1 dsa_ac_log_on" and "1.3.2 dsa_ac_log_off"
- new field in DSA audit records: the number of materialized entries in a search before

ACI is applied

- warnings in log files that server needs to be restarted after ACI changes
- detailed error codes and diagnostic messages in DSA audits
- CPU load measurements now being written to the DSA audit records
- new filter option -Q <query-ref> in dirxauddecode and dirxaudstatistics, see the "Administration Reference" manual, the sections "1.3 dirxauddecode" and "1.4 dirxaudstatistics" for detailed description of the -Q option and its usage.
- dirxaudstatistics summary extension and new top-lists for sub-durations
- logging of shadow agreements: history of SOB agreement operations helps ticket analysis in systems where replication is used.
- the "sob show" command of dirxadm application displays now more details about shadow agreements
- DBAM preload: extend the status information text of the 'in progress' status, which is accessed via the extended operation dsa_dbam_preload_status. Some more detail is returned about the currently processed items.
- improved log cleanup: not all the log files, generated by the DSA processes were deleted after time expiration, if DIRX_DEL_TIME was configured. Now all generated log files are handled equally. See the "Administration Reference" manual, section "4.2 DirX Directory Environment Variables" for description of DIRX_DEL_TIME.
- stack traces of crash dumps are now available on Windows as well as on Linux (in the past they were generated just on Linux). Refer to the ReadMe.txt file installed in the "crash" directory (on Windows) for more information.

DirX Manager:

- new DIRXADM node in the DSA section of Monitoring tab. Allows to execute dirxadm operations directly from the DirX Manager. More information on this feature can be found in the "Manager Guide", in the section "5.2.7 DSA dirxadm".

8.2.2. Discontinued Features

The LDAP Mib interface in dirxadm is deprecated. It will not be supported in future versions of DirX Directory. Use of the LDAP extended operations is recommended starting from version DirX Directory 8.4.

8.2.3. Changes to the User Interface or Configuration Defaults

Changes in the user interface or configuration since DirX Directory 8.9.

The configuration of crash dumps and stack traces has changed for Windows installation and must be reconfigured manually.Crash dumps are important for problems investigation by the DirX Directory support team.Please refer to the ReadMe_WIN.txt file installed in the "crash" directory (on Windows) for detailed instructions, in case you want to enable core dumps and stack traces on your system.

The specification of the serialNumber in the extensible match with the

CertificateExactMatchingRule or CertificateMatchingRule has changed.As of Dirx Directory V8.10 the value of the serialnumber has to be specified in hexadecimal notation.

No new SNMP traps since DirX Directory 8.9.

8.3. DirX Directory 8.9

8.3.1. New Features

Security: Support of version TLS1.3 throughout the whole product, i.e. in IDMS, ldapserver, dirxcp and DirX Manager.Look for the description of attribute supportedEncryptionStrengthExt in the Syntaxes and Attributes Manual and for the description of the environment variables DIRX_SET_TLS_LEVEL_MIN/MAX in this ReleaseNote document.

Enhanced DBAM Cache runtime checking and consistency control: Repair inconsistent DBAM Cache by an automatic reread of the affected DBAM pages from the DBAM device.

Extension to Dynamic Groups functionality: Support root as search base in ldap url of dynamic groups

Robustness: Support of the -repair option in db check also for subordinate index

Performance enhancements: Operation Support Perform Real Object Block Check in a multithreaded manner resulting in a shorter duration of "db check -rob" operation.

LDAP Proxy Extension: Search Result Rewrite Rules allow to specify a list the attributes that are to be returned. All other attributes returned from the target are skipped.

Diagnostics and logging: Several enhancements have been implemented to yield more diagnostic information,

- A performance profiling can enabled to trace performance issues.
- For better evaluation of search requests issues the search engine trace has been introduced. With this tool it is possible to trace the internal processing of the search engine.
- the information written to the DSA and LDAP audit recoreds has been extended and can be made visible using the triple -v options in dirxauddecode

sasl bind: extend SASL mapping certificate.extensions.altname.email to support configurable mapping attribute. Look for the description of the environment DIRX_MAP_CERT_ALTNAME_ATTR in this ReleaseNote document.

Linux installation: Linux installation does no longer use the ksh, furthermore a protocol is logged by the installation procedure.

Support of 2 additional proprietary LDAP Request Controls:

- LDAP_CTRL_COND_OP: Perform a modify or an add operation only if the ldap filter in the control value matches the target entry of the operation

- LDAP_CTRL_SEARCHRES_INFO: Return only statistics (number of entries, number of attributes and number of attrvalues) that a search operation would result in.

8.3.2. Discontinued Features

The LDAP Mib interface in dirxadm is deprecated.It will not be supported in future versions of DirX Directory.Use of the LDAP extended operations is recommended starting from version DirX Directory 8.4.

8.3.3. Changes to the User Interface or Configuration Defaults

No changes since DirX Directory 8.7. No new SNMP traps since DirX Directory 8.7.

9. Recommendations for an Operation Concept

Actions items which are strongly recommended to be included in an operations concept for DirX Directory are:

- Deploy the latest patch available for the product version installed.
- Periodically (e.g. daily), perform checks of disk space availability to make sure there is still enough space available for audit files, log files, backup files and LDIF dump files.
- Periodically (e.g. daily), perform physical backups of the DirX Directory database using the dirxbackup command. Check physical consistency of backup file with dirxbackup –T command.
- Periodically (e.g. daily), perform checks of database consistency using dbamverify command for backup files created with dirxbackup command. Risks can be minimized, if database inconsistencies are detected as early as possible. The availability of consistent backup files allows for fast recovery in case of failure.
- Perform dbamverify –ATXDS on the backup file to perform the consistency check.
- Periodically (e.g. weekly), perform checks of DBAM device capacity using the dbamdevinfo command. If fragmentation is larger than 60%, please perform defragmentation using the db purge –pob command. If space for entries or attribute indices is getting low, plan for a database extension in time.
- Periodically (e.g. monthly), create LDIF dump as logical backup of the DirX Directory Database using ldif_dump. Check the resulting ldif content file for completeness and errors (occurrence of string #ERROR:)
- Check for completeness of LDIF file: At the end of each LDIF file exported by the DSA a statistic is displayed. In order to check the completeness of the ldif export compare the # with the Number of processed records Note: Minor differences result from the fact that the number of processed entries exported by the DSA in the context of a ldif_dump or a create_total_ldif operation always includes the ROOT. The number of entries displayed by nmi show does not include the root and of GLUEs that might be located above the context prefix of the DIT.
- Perform emergency exercises regularly to ensure fast failure recovery.
- Look for DSA/LDAP server crashes. On Linux platforms look into crash directory. Every abnormal exit of the DirX Directory server processes leaves a crash file. On Windows platforms a crash results in an event being reported in the applications section of the Windows event viewer (Exception code: 0xc0000005, faulting application one of the DirX processes, e.g. dirxdsa.exe). In DirX Directory as of V8.2B there is an additional way to use the procdump tool in order to dump crashes into files. See the inline documentation of the variable DIRX_WDOG_[DSA|LDAP|PROGSVR]_DEBUG_MODE_ENABLED in the dirxenv.ini file.
- Report crashes to DirX Directory support team. For details of the commands mentioned above, please refer to the DirX Directory Administration Reference Guide.

For each of the commands listed above, we strongly recommend the following:

- Check the output of the commands carefully for error messages.
- Check the exit codes of the commands.

If you cannot fix the problem based on the information given with the error messages/exit codes, please inform the DirX Directory support team about the problem.

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.