LDAP/MSAD Plugin

Bacula Systems Documentation

Contents

Contents

1 Executive Summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership. This whitepaper presents solutions for object level backup and restore of Directory servers including Active Directory using the LDAP/MSAD Plugin with Bacula Enterprise version 10.

2 Overview

This document is intended to provide insight into the considerations and processes required to implement LDAP and MSAD Backup and Restore using **Bacula Enterprise**.

3 LDAP and MSAD

The LDAP Plugin was designed to perform a backup and restore of a single LDAP object. It uses the LDAP network protocol and the standard schema to search and fetch objects, so it should support a variety of different LDAP servers in addition to the OpenLDAP server which we have used for testing (for LDAP servers other than OpenLDAP, the plugin should be tested before using it in production).

The MSAD Plugin was designed to perform a backup and restore of a single MS Active Directory object. It uses the LDAP network protocol and Active Directory schema to search and restore objects. However, please be aware that even though it backs up Microsoft AD the plugin currently runs only on Linux and Unix machines. The MS Active Directory Plugin required the queries and code to be rewritten to correspond to the different Microsoft fields and structures. The MSAD Plugin was built and is run under Linux using a network connection to the Microsoft Active Directory server.

3.1 Features Summary

The **Bacula Enterprise** Directory Server Plugin provides the following main features for LDAP servers:

- support for backup levels: Full, Differential, Incremental
- support for Accurate mode (finds deleted objects)
- object size and modification time is properly saved in Bacula catalog
- optional object relocation during restore
- LDIF-like internal archive format
- supports replace options: always, never, ifnewer, ifolder
- connects to LDAP server using the LDAP network protocol
- support for the OpenLDAP server
- support for ldaps (SSL) communication with LDAP server
- support listing mode for browsing objects directly on Bacula console
- built and tested on Linux

In addition the MSAD Plugin provides the following additional features for Active Directory servers:

- support for Microsoft Active Directory server (from Windows 2003 up to the latest and future Windows versions)
- connects to MS Active Directory server using the LDAP network protocol
- allow automatic MS Active Directory objects tombstone recovery

- special Active Directory attributes handling

3.2 Scope

This paper presents solutions for **Bacula Enterprise** versions 10 and later, which are not applicable to prior versions.

4 Installation

The Bacula File Daemon and Directory Server Plugin can be installed on any Linux host which has access to the LDAP or MSAD server to be protected. Installation of the MSAD Plugin on a Windows Server is not currently supported.

4.1 Configuration

The **Plugin Directory** directive of the **File Daemon** resource in `/opt/bacula/etc/bacula-fd.conf` must point to where the `ldap-fd.so` or `msad-fd.so` plugin files are installed. The standard Bacula plugin directory is `/opt/bacula/plugins`

```
FileDaemon {
    Name = bacula-fd
    Plugin Directory = /opt/bacula/plugins
...
}
```

4.2 Installation of the Plugin

Installation of the Bacula Enterprise Directory Plugin is most easily done by adding the repository file suitable for the existing subscription to the Linux package manager for your distribution of choice. For RHEL/CentOS distributions an example would be `/etc/yum.respos.d/bee.repo` with the following content:

```
[bacula-enterprise]
name=Bacula Enterprise for RHEL $releasever - $basearch
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/10.0.1/rhel7-64
enabled=1
gpgcheck=1
gpgkey=https://www.baculasystems.com/dl/@customer-string@/BaculaSystems-Public-Signature.
↪asc

[bacula-enterprise-ldap]
name=Bacula Enterprise LDAP for RHEL $releasever - $basearch
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/ldap/10.0.1/rhel7-64
enabled=1
gpgcheck=1
gpgkey=https://www.baculasystems.com/dl/@customer-string@/BaculaSystems-Public-Signature.
↪asc
```

For Ubuntu/Debian distributions an example would be `/etc/apt/sources.list.d/bacula.list` with the following content:


```
#Bacula Enterprise
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/stretch-64/␣
↪stretch main
#Bacula Enterprise LDAP
deb https://www.baculasystems.com/dl/@customer-string@/debs/ldap/@version@/stretch-64/␣
↪stretch ldap
```

After that, a run of `apt-get update` for any Ubuntu/Debian distribution is needed. Then, the Plugin can be installed using

`apt-get install bacula-enterprise-ldap-plugin` or

`yum install bacula-enterprise-ldap-plugin` at RHEL/CentOS.

Manual installation of the packages, can be done after downloading the package files from the Bacula Systems provided download area, and then using the package manager to install.

5 Plugin Configuration

The plugin is configured using dedicated config files and/or **Plugin Parameters** defined in a Fileset **Include** section of the Bacula Enterprise Director configuration.

Note, even though you may be using the MSAD plugin to backup a Windows Active Directory, your File daemon will be a Linux or Unix machine. This limitation could change in the future.

5.1 Plugin Config file

The LDAP and MSAD plugins require a configuration file on the File Daemon machine. This configuration file contains parameters for LDAP or MSAD server connection. Each plugin uses its own configuration file.

The default configuration file is located at:

`/opt/bacula/etc/ldap.conf` for the LDAP plugin and

`/opt/bacula/etc/msad.conf` for the MSAD plugin.

You may specify a different config file (using `config=...` plugin parameter) in each FileSet definition to point your backup to different Directory servers.

You can see an example config file for LDAP Plugin:

```
#
# Sample config file for the ldap plugin /opt/bacula/etc/ldap.conf
#
LDAPURI = "ldap://192.168.0.100/"
BINDDN = "cn=backup,dc=acme,dc=com"
BINDPASS = "PASSW0RD"
BASEDN = "dc=acme,dc=com"
```

You can see an example config file for MSAD Plugin:


```
#
# Sample config file for the msad plugin
#
LDAPURI = "ldap://10.0.101.1/"
BINDDN = "cn=Administrator,cn=Users,dc=domain,dc=com"
BINDPASS = "password"
BASEDN = "dc=domain,dc=com"
```

The definitions and default values for Plugin config file parameters are described at Table *LDAP plugin parameters*. The config file supports the following parameters: `LDAPURI`, `BINDDN`, `BINDPASS`, `BASEDN`, `TLS_CACERT`, `TLS_CERFILE`, `TLS_CACERTDIR`, `TLS_REQCERT`. All other plugin parameters are not supported in the config file.

5.2 Plugin Parameters

The following plugin parameters can be used to configure connection, backup or restore jobs with a plugin.

Table 1: LDAP plugin parameters

Options	Default	Description
attribs		It is a comma separated list of additional attributes which plugin should skip during restore.
config	LDAP: `/opt/bacula/etc/ldap.conf` Active Directory: `/opt/bacula/etc/msad.conf`	The LDAP Plugin configuration file.
ldapuri	ldap://localhost/	The LDAP URI parameter specifies how to connect to the ldap server. The ldap, ldaps and ldapi schemes are supported.
binddn	cn=admin,dc=example,dc=com	Backup user distinguish name.
bindpass	secret	Backup user password.
hbindpass		Backup user password obscured.
basedn	dc=example,dc=com	A base location (DN) for backup, it could be a LDAP server's root tree or some other subtree.
TLS_CACERT	•	Path to a file on disk with the TLS Certificate
TLS_CACERTDIR	•	Path to a directory on disk with the TLS Certificates
TLS_REQCERT	demand	never, allow, try, demand
schemaapply	Yes	A boolean option (yes/no) to toggle an automatic skipping of well known read-only attributes of the Active Directory server (msad-fd plugin only).

Using the **config=...** plugin parameter, a prepared configuration file providing the plugin parameters can be used instead of the plugin command line.

Obscure the LDAP Password

As of the 8.0.3 version of the LDAP plugin, it is possible to obscure the password. The obscured password field is called **hpassword**. The bconsole `@encode` command can be used to generate the obscured password. Note that if the string to be encoded contains "=", the `string=` keyword with its parameter value in quotes has to be used:

```
# /opt/bacula/bin/bconsole
* @encode apassword
MTEyOjEyNzoGAwAYFQIVABEDAwcfAhQA

* @encode string="passwordwith="
NTMwOjU0Mzpic2FhZX1gdmV7ZnovAA
```

You can see an example config file for LDAP Plugin with user password obscured:

```
# cat /opt/bacula/etc/ldap.conf
#
# Sample config file for the ldap plugin /opt/bacula/etc/ldap.conf
#
LDAPURI = "ldap://192.168.0.100/"
BINDDN = "cn=backup,dc=acme,dc=com"
HBINDPASS = "NTMwOjU0Mzpic2FhZX1gdmV7ZnovAA"
BASEDN = "dc=acme,dc=com"
```

5.3 FileSet Examples

LDAP

In the example below, the plugin will use a default config file to make a backup:

```
FileSet {
    Name = FS_LDAPDefault
    Include {
        Plugin = "ldap:"
    }
}
```

In this example, the plugin will use a selected config file to make a backup:

```
FileSet {
    Name = FS_LDAPConfig
    Include {
        Plugin = "ldap: config=/opt/bacula/etc/openldap-server.conf"
    }
}
```

In the following example all required config parameters are provided at the Plugin=... command line. No config file is used:

```
FileSet {
    Name = FS_LDAPAll
    Include {
        Plugin = "ldap: ldapuri=ldap://192.168.0.200/ binddn=DOM/Administrator \
```

(continues on next page)

(continued from previous page)

```
            bindpass=secret basedn=dc=dom,dc=com"
    }
}
```

MSAD

In the example below, the plugin will use a default config file to make a backup:

```
FileSet {
    Name = FS_MSADDefault
    Include {
        Plugin = "msad:"
    }
}
```

In this example, the plugin will use a selected config file to make a backup:

```
FileSet {
    Name = FS_MSADConfig
    Include {
        Plugin = "msad: config=/opt/bacula/etc/msad-server.conf"
    }
}
```

In the following example all required config parameters are provided at the Plugin=... command line. No config file is used:

```
FileSet {
    Name = FS_MSADAll
    Include {
        Plugin = "msad: ldapuri=ldap://10.0.101.1/ binddn=cn=Administrator,cn=Users,
↪dc=domain,dc=com \
            bindpass=password basedn=dc=domain,dc=com"
    }
}
```

6 Preparation

You may specify a different config file in each FileSet definition. Note that the configuration file is loaded by the plugin, not the Director, thus it needs to be located on the host running the File Daemon. The plugin requires a LDAP/MSAD account with permissions to query and read objects for backup. This account can be an admin account or standard account with a Backup Operator role. LDAP/MSAD plugin creates a virtual namespace in Bacula catalog which consists of “ldap:” or “msad:” prefixes and the DIT is represented as a directory tree. The Bacula virtual namespace does not contain the ldap/msad server/instance name so the backup admin must distinguish the same tree between different servers on the same Bacula FD client.

6.1 Testing the Connection

To test the connection parameters for your LDAP or MSAD server, you can use the following command:

```
# /opt/bacula/bin/ldaptest
Usage: ./ldaptest <uri> <binddn> <bindpass>
```

where:

uri is the value set in `LDAPURI` config parameter

binddn is the value set in `BINDDN` config parameter

bindpass is the value set in `BINDPASS` config parameter

You can find an example of `ldaptest` utility execution below which display ldap debug information during connection initiation and server Root DSE information on successful connection.

```
# /opt/bacula/bin/ldaptest ldap://192.168.0.200/ 'DOM\Administrator' password
ldap_sasl_bind_s
ldap_sasl_bind
ldap_send_initial_request
ldap_new_connection 1 1 0
ldap_int_open_connection
ldap_connect_to_host: TCP 192.168.0.200:389
ldap_new_socket: 3
ldap_prepare_socket: 3
ldap_connect_to_host: Trying 192.168.0.200:389
ldap_pvt_connect: fd: 3 tm: -1 async: 0
attempting to connect:
connect success
ldap_open_defconn: successful
ldap_send_server_request
ldap_result ld 0x55cbc7b57630 msgid 1
wait4msg ld 0x55cbc7b57630 msgid 1 (infinite timeout)
wait4msg continue ld 0x55cbc7b57630 msgid 1 all 1
** ld 0x55cbc7b57630 Connections:
* host: 192.168.0.200  port: 389  (default)
refcnt: 2  status: Connected
last used: Mon May 21 08:48:59 2018
(...)
LDAP Server connection OK

RootDSE:
    currentTime: 20180521154906.0Z
    subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=dom,DC=com
    namingContexts: DC=dom,DC=com
    namingContexts: CN=Configuration,DC=dom,DC=com
    namingContexts: CN=Schema,CN=Configuration,DC=dom,DC=com
    namingContexts: DC=DomainDnsZones,DC=dom,DC=com
    namingContexts: DC=ForestDnsZones,DC=dom,DC=com
    defaultNamingContext: DC=dom,DC=com
    schemaNamingContext: CN=Schema,CN=Configuration,DC=dom,DC=com
    configurationNamingContext: CN=Configuration,DC=dom,DC=com
    rootDomainNamingContext: DC=dom,DC=com
```

(continues on next page)

(continued from previous page)

```
(...)
Test successful!
```

To find a BINDDN for your user for Active Directory you can use a PowerShell command below:

```
PS> dsquery user -name Administrator
"CN=Administrator,CN=Users,DC=dom,DC=com"
```

7 Backup

The plugin performs a single base query to find all objects to backup. The plugin will use a paged control response to get all available objects. It saves all standard and extended attributes returned from server. This includes any system or dynamic attributes.

When the plugin detects a referral object during backup, it will not descend to it and will log a message, i. e.:

```
(...) Referal found. Will not descend to ref: ldap://.../...
```

The backup will create a single file for every object found in Directory Server subtree started from `BASEDN` parameter.

7.1 Testing your FileSet

The `estimate` command can be used to test your `FileSet`, especially with the `listing` parameter.

```
* estimate listing job=pluginTest level=Full
Using Catalog "MyCatalog"
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
drwxr-xr-x  1 root  root   585 2014-03-25 10:12:22  ldap:/dc=com/dc=bacula/
-rw-r--r--  1 root  root   542 2014-03-25 10:12:22  ldap:/dc=com/dc=bacula/cn=root
-rw-r--r--  1 root  root   535 2014-03-25 10:12:22  ldap:/dc=com/dc=bacula/cn=test
2000 OK estimate files=3 bytes=1,077
```

8 Restore

The plugin supports restore operations to a working LDAP or MSAD server only. You cannot restore objects to the local filesystem. You can however restore objects to their original location or relocate them to different parts of the subtree.

To restore an object to its original location you have to use a **where=/** restore parameter.

The Plugin is not designed for Disaster Recovery procedures where the LDAP or MSAD servers may not be functional. For Disaster Recovery of an MS AD servers you can use the Bacula Enterprise VSS plugin.

LDAP objects are restored like regular files with the Bacula “restore” command.

```
cwd is: /
$ cd ldap:/dc=com/dc=bacula
cwd is: ldap:/dc=com/dc=bacula/
$ dir
```

(continues on next page)

(continued from previous page)

```
-rw-r--r-- 1 root  root  568    ldap:/dc=com/dc=bacula/cn=admin
drwxr-xr-x 1 root  root  480    ldap:/dc=com/dc=bacula/ou=Accounting/
drwxr-xr-x 1 root  root  491    ldap:/dc=com/dc=bacula/ou=Administrative/
drwxr-xr-x 1 root  root  494    ldap:/dc=com/dc=bacula/ou=Human Resources/
drwxr-xr-x 1 root  root  479    ldap:/dc=com/dc=bacula/ou=Janitorial/
drwxr-xr-x 1 root  root  479    ldap:/dc=com/dc=bacula/ou=Management/
drwxr-xr-x 1 root  root  470    ldap:/dc=com/dc=bacula/ou=Payroll/
drwxr-xr-x 1 root  root  464    ldap:/dc=com/dc=bacula/ou=Peons/
drwxr-xr-x 1 root  root  506    ldap:/dc=com/dc=bacula/ou=Product Development/
drwxr-xr-x 1 root  root  494    ldap:/dc=com/dc=bacula/ou=Product Testing/
drwxr-xr-x 1 root  root  450    ldap:/dc=com/dc=bacula/ou=groups/
drwxr-xr-x 1 root  root  448    ldap:/dc=com/dc=bacula/ou=hosts/
drwxr-xr-x 1 root  root  450    ldap:/dc=com/dc=bacula/ou=people/
$ add "ou=Product Testing"
121 files marked.
$ cd ou=people
cwd is: ldap:/dc=com/dc=bacula/ou=people/
$ dir
-rw-r--r-- 1 root  root 1006618   ldap:/dc=com/dc=bacula/ou=people/uid=john
$ add *
1 file marked.
```

or for the MSAD plugin:

```
$ dir
drwxr-xr-x 1 root  root   924    msad:/DC=com/DC=bacula/DC=msad/CN=Builtin/
drwxr-xr-x 1 root  root   621    msad:/DC=com/DC=bacula/DC=msad/CN=Computers/
-rw-r--r-- 1 root  root   723    msad:/DC=com/DC=bacula/DC=msad/CN=Infrastructure
-rw-r--r-- 1 root  root   630    msad:/DC=com/DC=bacula/DC=msad/CN=LostAndFound
-rw-r--r-- 1 root  root   663    msad:/DC=com/DC=bacula/DC=msad/CN=NTDS Quotas
drwxr-xr-x 1 root  root   579    msad:/DC=com/DC=bacula/DC=msad/CN=Program Data/
drwxr-xr-x 1 root  root   583    msad:/DC=com/DC=bacula/DC=msad/CN=System/
drwxr-xr-x 1 root  root   601    msad:/DC=com/DC=bacula/DC=msad/CN=Users/
-rw-r--r-- 1 root  root  1495    msad:/DC=com/DC=bacula/DC=msad/CN=backup
drwxr-xr-x 1 root  root   775    msad:/DC=com/DC=bacula/DC=msad/OU=Domain Controllers/
$ add CN=backup
1 file marked.
```

You can change the restore subtree using a **where=...** parameter with restore command. It should contain a relocation DN, i.e:

```
* restore where = "dc=restore,dc=example,dc=com"
```

The LDAP or MSAD Plugin relocation restore works similar to a standard **where=/tmp/restores** does for regular files and directories recovering the whole object's subtree path. You may also change the replace mode during the restoration process. The supported modes are: *always*, *never*, *ifnewer*, *ifolder*. Replace mode can be changed during `bconsole restore` command execution.

The plugin saves all standard and extended attributes even if they are not directly recoverable or recoverable at all (read only attributes). Some of the saved attributes will be indirectly restored (like **memberOf** attribute), others will be simply skipped during restore. In this case the you will get appropriate information in the job log, similar to the following example below:


```
JobId 220: msad: Restore for AD Schema(69): Windows Server 2012 R2
JobId 220: Elapsed time=00:00:01, Transfer rate=1.021 K Bytes/second
JobId 220: msad: Skipping read-only attribute: DN: ... Attr: whenCreated
JobId 220: msad: Skipping read-only attribute: DN: ... Attr: whenChanged
(...)
```

8.1 Restore Options

Both plugins support a restore options interface where you can set required plugin variables. With restore options interface you can change the location of the restore server, bind user or specific restore mechanism.

To use the restore options interface you need to select option 13 of the modification interface during restore process:

```
(...)
13: Plugin Options
Select parameter to modify (1-13): 13
```

For LDAP Plugin you have a following options:

```
Plugin Restore Options
basedn:              *None*              ()
ldapuri:             *None*              ()
binddn:              *None*              ()
bindpass:            *None*              ()
hbindpass:           *None*              ()
config:              *None*              ()
attribs:             *None*              ()

You have the following choices:
1: basedn (Base location (DN) for restore, it could be a ldap server root tree \
    or some other subtree. ex: dc=example,dc=com)
2: ldapuri (Universal Resource Identifier for MSAD server, connection string. \
    ex: ldap://192.168.0.100/)
3: binddn (Restore user distinguish name. ex: cn=backup,dc=example,dc=com)
4: bindpass (Restore user password)
5: hbindpass (Restore user password obscured)
6: config (Path to alternate configuration file)
7: attribs (Additional attribute list to skip during the restore (comma separated list))
```

And for MSAD Plugin you have the following options:

```
Plugin Restore Options
basedn:              *None*              ()
ldapuri:             *None*              ()
binddn:              *None*              ()
bindpass:            *None*              ()
hbindpass:           *None*              ()
config:              *None*              ()
schemaapply:         *None*              (yes)
attribs:             *None*              ()

You have the following choices:
```

(continues on next page)

(continued from previous page)

```
1: basedn (Base location (DN) for restore, it could be a ldap server root tree \
    or some other subtree. ex: dc=example,dc=com)
2: ldapuri (Universal Resource Identifier for MSAD server, connection string. \
    ex: ldap://192.168.0.100/)
3: binddn (Restore user distinguish name. ex: cn=backup,dc=example,dc=com)
4: bindpass (Restore user password)
5: hbindpass (Restore user password obscured)
6: config (Path to alternate configuration file)
7: schemaapply (Skip automatically well known read-only attributes)
8: attribs (Additional attribute list to skip during the restore (comma separated list))
```

- **ldapuri** Universal Resource Identifier for LDAP/MSAD server, connection string
- **binddn** backup user distinguish name
- **bindpass** backup user password
- **hbindpass** backup user encoded password
- **basedn** A base location (DN) for backup, it could be a ldap server root tree or some other subtree
- **config** A config file to use which has all required parameters
- **schemaapply** A boolean option (yes/no) to toggle an automatic skipping of well known read-only attributes of the Active Directory server
- **attribs** It is a comma separated list of additional attributes which plugin should skip during restore

When you restore LDAP or Active Directory objects sometimes you can get a following error:

```
ldap err: 53 errmsg: "0000209A: SvcErr: DSID-031A104A, problem 5003 (WILL_NOT_PERFORM)
```

which means some of the restored attributes cannot be restored due to restrictions set at Directory server side. These attributes need to be skipped during restore to meet the restrictions. The plugin can automatically skip some well known system/read-only attributes when you toggle the restore parameter: **schemaapply** to yes (the default value). Then during restore a plugin will verify the Active Directory schema number and apply the correct restore filter to this attributes. You can extend this filter with your own list using **attribs** restore parameter. This parameter is available for LDAP and MSAD plugin. The **schemaapply** parameter is available for MSAD plugin only.

The restore job log will provide information about attributes skipped during restore, see *Restore*.

8.2 Active Directory restore

Active Directory is a special kind of Directory server which requires special treatment during restore operations, and this is handled by the dedicated MSAD plugin.

Tombstone recovery

Note: New in 10.0.2

As of version 10.0.2, the MSAD Plugin supports Active Directory tombstone objects recovery.During restore the MSAD Plugin automatically checks if a restored object has an available tombstone. If found the MSAD Plugin recovers it from tombstone and restores all other attributes from backup. This feature allows proper recovery for some system attributes like `SID` or `objectGUID` which aren't recoverable with other methods.

memberOf recovery

The plugin recognizes the `memberOf` object's attribute during restore and handles it correctly. After a successful object restore it adds restored object to all groups pointed by the `memberOf` attribute list. When a destination group does not exist on the AD server this group will be skipped.

userAccountControl recovery

The `userAccountControl` attribute in Active Directory requires special permissions for successful restore. It may be required to add `Enterprise Admins` security group to your `BINDDN` restore user or modify `BINDDN` user permissions directly.

The plugin will check if `userAccountControl` attribute restore is allowed with current permissions and inform the user in the job log if the restore was unsuccessful. In this case `userAccountControl` attribute will be restored with default flags, i. e. user account will be disabled.

sAMAccountName

The `sAMAccountName` attribute found in Active Directory user object is a special attribute which has to be unique on the whole directory context. This requires special handling during a relocation restore when an original user object already exist in its original location. In this case the object relocated during restore will be altered. The default `sAMAccountName` attribute will be suffixed with _$(JobID), the value of the restore job, i. e. when an original `sAMAccountName` attribute has a value: `testuser` then during restore it becomes: `testuser_230` where the restore job id is 230. This `sAMAccountName` attribute change is performed only when the following conditions are met:

- restore job executed with relocation - **where=...** parameter different from `/`
- the `sAMAccountName` attribute value already exists on directory context, i. e. an original user object still exists on directory server

9 Other

9.1 Common Problems

- Error: msad plugin: ldap err: 50 errmsg: "00000005: SecErr: DSID-031521E1, problem 4003 (INSUFF_ACCESS_RIGHTS) - Your backup and restore user has no rights to create or modify restored objects, please add a required permissions to the user.
- Error: msad plugin: ldap err: 53 errmsg: "0000209A: SvcErr: DSID-031A104A, problem 5003 (WILL_NOT_PERFORM), data 0 - Some of restored AD attributes cannot be modified and you need to exclude them using Restore Plugin Options as described in **restoreopt**.
- You have to use `ldap:` or `msad:` as a name of the plugin (please check the colon ':' character) even when no additional parameters were set. Without it the job won't backup any objects and finishes without an error.

9.2 Object listing

The Bacula Enterprise LDAP/MSAD Plugin supports the new Plugin Listing feature of Bacula Enterprise 8.x or newer. This mode allows a Plugin to display some useful information about available LDAP/MSAD objects or server default name context.

The new feature uses the special **.ls** command with a new **plugin=<plugin>** parameter. The command requires the following parameters to be set:

client=<client> A Bacula Client name with the LDAP/MSAD Plugin installed.

plugin=<plugin> A Plugin name, which should be **ldap:** or **msad:** in this case, with optional plugin parameters as described in section **parameters**.

path=<path> An object path to display.

If you do not specify a plugin parameters in command line then plugin will use parameters defined in default config file (i.e. `/opt/bacula/etc/ldap.conf`) or default parameters value which in most cases are very different from your environment. All examples below assumes you have a valid default config file available.

The supported values for a **path=<path>** parameter are:

`/` to display ldap server default name context.

`DN` to display all child objects at a following **DN**.

To display ldap server default name context, use the following command example:

```
*.ls client=bacula-fd plugin=msad: path=/
Connecting to Client bacula-devel-fd at bacula-devel:9102
drwxr-x---   1 root     root               0 2018-05-21 09:31:39  DC=dom,DC=com
2000 OK estimate files=1 bytes=0
```

To display the contents of a selected ldap subtree :

```
*.ls client=bacula-devel-fd plugin=msad: path=DC=dom,DC=com
Connecting to Client bacula-devel-fd at bacula-devel:9102
 root     root     137 2018-05-04 20:37:32  CN=Builtin,DC=dom,DC=com/
 root     root     135 2018-05-04 20:37:28  CN=Computers,DC=dom,DC=com/
 root     root     153 2018-05-04 20:37:29  OU=Domain Controllers,DC=dom,DC=com/
 root     root     151 2018-05-04 20:37:29  CN=ForeignSecurityPrincipals,DC=dom,DC=com/
 root     root     151 2018-05-04 20:37:29  CN=Infrastructure,DC=dom,DC=com
```

(continues on next page)

(continued from previous page)

```
 root     root     141 2018-05-04 20:37:28  CN=LostAndFound,DC=dom,DC=com
 root     root     150 2018-05-04 20:37:29  CN=Managed Service Accounts,DC=dom,DC=com/
 root     root     147 2018-05-04 20:37:29  CN=NTDS Quotas,DC=dom,DC=com
 root     root     138 2018-05-04 20:37:29  CN=Program Data,DC=dom,DC=com/
 root     root     142 2018-05-16 21:47:46  OU=restore,DC=dom,DC=com/
 root     root     132 2018-05-04 20:37:28  CN=System,DC=dom,DC=com/
 root     root     161 2018-05-04 20:37:29  CN=TPM Devices,DC=dom,DC=com
 root     root     131 2018-05-04 20:37:28  CN=Users,DC=dom,DC=com/
2000 OK estimate files=13 bytes=600
```

9.3 Limitations

- For a restore to work, the LDAP/MSAD server must be fully functional.
- It is not possible to restore LDAP objects as a regular files because LDAP nodes (bacula directories in catalog) contain attributes (data to restore) like LDAP leafs (bacula files in catalog). These limitations may be removed in the future.
- Active Directory tombstone recovery is performed automatically for the MSAD Plugin and cannot be disabled.
- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

