Directory Server 1

Red Hat Directory Server 8.1 Release Notes

for Directory Server 8.1

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588 Research Triangle Park, NC 27709 USA

These release notes contain important information available at the release of Red Hat Directory Server version 8.1. New features, system requirements, installation notes, known problems, resources, and other current issues are addressed here. Read this document before beginning to use Directory Server 8.1.

1. New in Red Hat Directory Server 8.1

Directory Server 8.1 has introduced many features to make managing the directory service and its data easier.

1.1. Enhanced Server to Server Connections with Added SASL/Di-

gest-MD5 (Kerberos), SASL/GSSAPI (Kerberos), and Start TLS Support

Red Hat Directory Server performs a number of different connections between servers, such as replication, chaining, synchronization, and pass-through authentication. To secure these connections, Red Hat Directory Server previously supported SSL and TLS authentication. Directory Server 8.1 expands the secure connection options to include SASL Kerberos and Start TLS for these server to server operations.

Connections between Directory Server instances can be secured using SASL Kerberos and Start TLS. This includes replication and chaining (database links).

Pass-through authentication now allows optional arguments to enable Start TLS. (SASL connections are not supported for pass-through authentication.)

Windows synchronization now supports Start TLS (a secure TLS connection over a standard LDAP port) for Active Directory-Directory Server connections. (SASL connections are not supported for Windows.)

The configuration attributes and Console has been updated to include these enhancements:

- For replication and synchronization, the *nsds5ReplicaBindMethod* and *nsds5ReplicaTransportInfo* attributes

- For chaining, the *nsUseStartTLS*, *nsBindMechanism* and *nsActiveChainingComponents* attributes

1.2. Moved Server Task Management to LDAP with cn=tasks Entries

Directory Server 8.1 has the ability to launch server maintenance tasks over LDAP. Directory tasks like import, export, backup, restore, and indexing, as well as new tasks for reloading schema and updating people's group membership attributes.

Each task has its own entry under the **cn=tasks,cn=config** configuration entry in the server's DSE. A new task entry can be added, with task-specific attributes, to initiate the task. As soon as the task is completed, the task entry is removed. For example, this launches a task to create a new index:

```
/usr/lib/mozldap/ldapmodify -a -D "cn=directory manager" -w secret -p 389 -
h server.example.comdn: cn=example presence index, cn=index, cn=tasks,
cn=configobjectclass: nsDirectoryServerTaskcn: example presence indexnsIn\
dexAttribute: "cn:pres"
```

All seven tasks and their allowed attributes are covered in the **cn=tasks,cn=config** section of the core configuration chapter in the *Red Hat Directory Server Configuration, Command, and File Reference*.

1.3. Improved Schema Extensions through Dynamic Schema Reloads

Previous to Red Hat Directory Server 8.1, if custom schema file was added to the Directory Server, the Directory Server instance had to be restarted to load the schema.

Directory Server 8.1 introduces a dynamic schema reload task, which allows custom schema files to be added to an instance and loaded on the fly. This simplifies extending Directory Server schema.

Dynamic schema reload is supported through the **`cn=tasks,cn=config`** entry (by adding a task entry beneath the **`cn=schema reload task, cn=tasks,cn=config`** container entry) and through a new script, **`schema-reload.pl`**.

1.4. Added Support for Unix Sockets and Autobind

Inter-process communication (IPC) is a way for processes on a Unix machine or network to communicate directly with each other. Running LDAP operations over IPC connections is called *LDAPI*. Directory Server 8.1 introduces LDAPI support, meaning that Directory Server's LDAP operations can run over Unix sockets.

Enabling LDAPI also allows the Directory Server to use autobind to authenticate logged in Unix users to the Directory Server automatically, based on their Unix credentials.

Both LDAPI and autobind are configured through new core configuration attributes which have been added to the Directory Server.

1.5. Added New Plug-in for Automatically Managing and Assigning Numbers for Attributes

Some entry attributes require having a unique number, such as *`uidNumber`* and *`gidNumber`*. Directory Server 8.1 introduces the *Distributed Numeric Assignment (DNA) Plug-in*. This plug-in assigns ranges of numbers to a server, and the server then assigns numbers to attributes, based on their subtree and a matching filter. Ranges can be reallocated among supplier servers to make sure that a server always has an adequate range without assigning duplicate numbers.

1.6. Added New Plug-in to Simplify Group Membership Management

Group membership is defined in the group entry itself. For static groups, members are identified by their DN in the *`member`* or *`uniqueMember`* attribute. However, before Directory Server 8.1, there was no way to tell by looking at a user entry what groups the user was a member of.

Directory Server 8.1 has added a new managed attribute, *`memberOf`*, and a new MemberOf Plug-in. Whenever a member is added to a static group, the MemberOf Plug-in uses the person's DN from the *`member`* or *`uniqueMember`* attribute to search for the user entry, and then adds a *`memberOf`* attribute to the user entry. This way, it's simple to tell from looking at the user entry what groups it belongs to.

`memberOf` attributes are initially assigned to entries by running a special task. This task can be launched by creating a task entry beneath the **`cn=memberof task, cn=task,cn=config`** container entry or by running the new **`fixup-memberof.pl`** script.

1.7. Extended Get Effective Rights Operations with Options for Non-Existent and Operational Attributes

A *get effective rights* operation is an extended **`ldapsearch`** that, along with regular search results, returns that access permissions that one directory user has to a directory entry or entries.

Directory Server 8.1 adds two additional attribute search options for get effective rights searches. One (**`*`**) returns rights for non-existent attributes for the entry, meaning attributes which could be set on the entry but currently are not. The other (**`+`**) returns access rights for operational attributes for the entry.

1.8. Added New Support for 64-Bit Integers for Performance Counters on 32-Bit Systems

Many of the performance counters for the Directory Server — including server statistics, database statistics, and SNMP monitoring — record 32-bit integers. For large or high-traffic systems, these counters may roll over too quickly, creating quirky performance statistics and making it difficult to conduct long-term analysis.

Directory Server 8.1 introduces support for 64-bit integers for performance counters, even on 32-bit systems. These 64-bit integers are enabled through a new configuration attribute on the DSE, *`nsslapd-counters`*. When 64-bit integers are enabled, all available counters support 64-bit integers.

For server statistics, there are five counters which support 64-bit integers:

- opsinitiated
- opscompleted
- entriessent
- bytessent
- totalconnections

For database statistics, there are four counters which support 64-bit integers:

- entrycachehits
- entrycachetries
- currententrycachesize
- maxentrycachesize

All of the attributes monitored by SNMP can support 64-bit integers.

1.9. Added New Parameter for Setting the Interval for Win Sync Checks

In synchronization, updates are sent two ways, from the Directory Server to the Active Directory server and from Active Directory back to the Directory Server. The frequency which Directory Server sends updates to Active Directory is set in the synchronization schedule, handled by the *nsds5replicaupdateschedule* attribute. The frequency which Directory Server checked Active Directory for updates was hard coded at five minutes.

A new attribute has been added, *winSyncInterval*, which sets how frequently the Directory Server should check the Active Directory peer for changes. If this attribute is not set, the default frequency is still every five minutes.

This new Win Sync interval can be used with existing sync agreements. To apply this new attribute:

1Upgrade the software, as described in *Section 3.4, “Upgrading to Directory Server 8.1”*.

2Copy the **01common.ldif** from the common **/etc/dirsrv/schema** directory into the instance-specific directory, such as **/usr/lib/dirsrv/slapd-*instance_name*/schema**.

It is okay to overwrite the new **01common.ldif** schema file because it is new and because the core configuration schema should never be modified, so there shouldn't be any custom settings.

3Reload the schema. For example:

```
/usr/lib/dirsrv/slapd-instance_name/schema-reload.pl -D "cn=Directory Man\
ager" -w secret
```

4Edit the sync agreement to add the *winSyncInterval* attribute.

```
/usr/lib/mozldap/ldapmodify -a -D "cn=directory manager" -w secret -p 389 -
h server.example.comdn: cn=ExampleSyncAgreement,cn=sync rep\
lica,cn="dc=example,dc=com",cn=mapping tree,cn=configchangetype: modifyadd:
winSyncIntervalwinSyncInterval: 600
```

1.10. Added a New Parameter to Control How the Server Handles Unauthenticated Binds

Users can attempt to bind to the directory using a username but without giving a password. For example, this command does not include the -w option or any other password option:

```
/usr/lib/mozldap/ldapsearch -D "cn=directory manager" -b
"dc=example,dc=com" -s sub "(objectclass=*)"
```

This is called an *unauthenticated bind*, because the user as whom to bind is given, but without any credentials.

Before 8.1, the Directory Server allowed that unauthenticated bind to continue as an anonymous bind. However, this created a management issue for servers which did not allow anonymous binds and a security risk for ones which did.

A new configuration attribute, *`nsslapd-allow-unauthenticated-binds`*, sets whether to allow an unauthenticated bind to succeed as an anonymous bind or whether the bind attempt fails. By default, this is turned off, so that unauthenticated binds fail, which is more secure.

```
nsslapd-allow-unauthenticated-binds: off
```

2. System Requirements

This section contains information related to installing and upgrading Red Hat Directory Server 8.1, including prerequisites and hardware or platform requirements.

2.1. Required JDK

Red Hat Directory Server 8.1 requires Sun JRE 1.6.0 or OpenJDK 1.6.0 for Red Hat Enterprise Linux 5 and HP-UX.

> **IMPORTANT**
>
> When the new JDK is installed for Directory Server 8.1, it is no longer possible to manage older instances of Directory Server using the Directory Server Console because the required JDKs for the different Directory Server versions are different. You must migrate any older instance to Directory Server 8.1 if you need to manage that instance with the Directory Server Console.

Red Hat Directory Server 8.1 requires Java IBM 1.6.0 for Red Hat Enterprise Linux 4.

2.2. Perl Prerequisites

Directory Server 8.1 does not package **nsperl** with the product. **perldap** should work with the version of **perl** pre-installed on the system.

There are some prerequisites for **perl** to run **perldap** with the pre-installed version.

- For Red Hat Enterprise Linux systems, use the Perl version that is installed with the operating system in **`/usr/bin/perl`** for both 32-bit and 64-bit versions of Red Hat Directory Server.

- On Solaris systems, Red Hat Directory Server is installed with a Perl package, **`RHATperlx`**, that must be used. This package contains a 64-bit version of Perl 5.8. It is not possible to use the Perl version installed in **`/usr/bin/perl`** on Solaris because it is 32 bit and will not work with Directory Server's 64-bit components.

- On HP-UX, Red Hat Directory Server uses the Perl version installed with the operating system in `/opt/perl_64/bin/perl`. Contact Hewlett-Packard support if this Perl version is not installed.

2.3. Directory Server Supported Platforms

Directory Server 8.1 is supported on the following platforms:

- HP-UX 11i Itanium/IPF
- Red Hat Enterprise Linux 4 i386 (32-bit)
- Red Hat Enterprise Linux 4 x86_64 (64-bit)
- Red Hat Enterprise Linux 5 i386 (32-bit)
- Red Hat Enterprise Linux 5 x86_64 (64-bit)

> **NOTE**
>
> Red Hat Directory Server 8.1 is supported running on a virtual guest on a Red Hat Enterprise Linux 5 virtual server.

- Sun Solaris 9 (SPARC v9, 64-bit)

2.4. Directory Server Console Supported Platforms

The Directory Server Console is supported on the following platforms:

- HP-UX 11i Itanium/IPF
- Red Hat Enterprise Linux 4 i386 (32-bit)
- Red Hat Enterprise Linux 4 x86_64 (64-bit)
- Red Hat Enterprise Linux 5 i386 (32-bit)
- Red Hat Enterprise Linux 5 x86_64 (64-bit)

- Sun Solaris 9 (SPARC v9, 64-bit)
- Windows XP
- Windows 2000 Server
- Windows 2003 Server

> **NOTE**
>
> The Directory Server Console can be installed on additional Windows platforms at an additional cost.

2.5. Windows Sync Service Platforms

The Windows Sync tool runs on these Windows platforms:

- Windows 2003 Active Directory (32-bit)
- Windows 2000 Active Directory (32-bit)

2.6. Web Application Browser Support

Directory Server 8.1 supports the following browsers to access web-based interfaces, such as **Admin Express** and online help tools for administrators and **Org Chart** and **Phonebook** for all users:

- Firefox 1.0 (Red Hat Enterprise Linux 4 and Solaris 9)
- Mozilla 1.4 (HP-UX)
- Mozilla 1.4.3 (Solaris 9)
- Mozilla 1.7.3 (Red Hat Enterprise Linux 4)
- Microsoft Internet Explorer 6.0 (Windows)

NOTE

Red Hat Directory Server web tools like Admin Express and Org Chart are not supported on Netscape browsers or any browser running on Mac.

3. Installing Directory Server 8.1

For more detailed instructions on installing Directory Server 8.1, see the *Directory Server Installation Guide* at http://www.redhat.com/docs/manuals/dir-server/.

3.1. Installing the JDK

Directory Server 8.1 requires Sun JRE 1.6.0 or OpenJDK 1.6.0. The appropriate Sun JDK should already be available on Sun Solaris systems, but it is necessary to install the JDK separately for other platforms. Either Sun JDK 6.0 or OpenJDK 1.6.0 is allowed.

For example, to install OpenJDK on Red Hat Enterprise Linux 5:

```
yum install java-1.6.0-openjdk
```

OpenJDK is also available for download from *http://openjdk.java.net/install/* for Red Hat Enterprise Linux and HP-UX.

For Red Hat Enterprise Linux 4, subscribe to the Extras channel in Red Hat Network, and install Java IBM 1.6.0 using **`up2date`**:

```
up2date java-1.6.0-ibm
```

IMPORTANT

When the new JDK is installed for Directory Server 8.1, it is no longer possible to manage older instances of Directory Server using the Directory Server Console because the required JDKs for the different Directory Server versions are different. You must migrate any older instance to Directory Server 8.1 if you need to manage that instance with the Directory Server Console.

3.2. Obtaining Packages

Red Hat Directory Server 8.1 packages are available for download from Red Hat Network (*http://rhn.redhat.com*). Downloading packages from Red Hat Network requires specific entitlements for the account for the 8.1 release.

To download Red Hat Directory Server 8.1 packages, log into Red Hat Network, then open the Red

Hat Directory Server 8.1 channel in **Channels** and go to the **Downloads** tab.

Both RPMs and ISO images are available for download through Red Hat Network. RPM packages can be downloaded and installed using **`rpm`**. The ISO images for Red Hat Enterprise Linuxand Solaris can be downloaded and burned on to a CD-recordable media using the appropriate software.

Along with the packages, there are tarball (**`.tar.gz`** file) archives for the source code.

> **NOTE**
>
> The source files are tarball (**`.tar.gz`**) archive files, not ISO images.

Red Hat Enterprise Linux customers can use Red Hat Network to obtain packages, or they can simply install or update their packages using **`yum`** or **`up2date`**, using an account with entitlements for the Red Hat Directory Server 8.1 release.

> **IMPORTANT**
>
> Although the Win Sync packages are listed in every Directory Server channel in Red Hat Network (Solaris, Red Hat Enterprise Linux 32-bit and Red Hat Enterprise Linux 64-bit), Win Sync is only supported on 32-bit Windows machines.

Directory Server packages are installed using native package management tools. For example, on Red Hat Enterprise Linux:

```
ls *.rpm | egrep -iv -e devel -e debuginfo | xargs rpm -ivh
```

On Sun Solaris:

```
for pkg in *.pkg ; do    pkgadd -d $pkg alldone
```

3.3. Running setup-ds-admin.pl

After installing the packages, run the **`setup-ds-admin.pl`** script to configure the new Directory Server and Administration Server instances. For example:

```
setup-ds-admin.pl
```

See the *Directory Server Installation Guide* for more information about **`setup-ds-admin.pl`** script options and the Directory Server configuration interface.

3.4. Upgrading to Directory Server 8.1

Red Hat Enterprise Linux systems support an in-place upgrade when moving from Red Hat Directory Server 8.0 to Red Hat Directory Server 8.1. To do this, install or update the RPMs, then re-run the setup script with the `-u` option.

```
setup-ds-admin.pl -u
```

This updates the settings automatically, without having to migrate or re-configure the server.

3.5. Migrating to Directory Server 8.1

Upgrading from Red Hat Directory Server 7.1 to Directory Server 8.1 requires a migration. The migration process has a special script, **`migrate-ds-admin.pl`**, which copies the data and configuration from the 7.1 instance to the new 8.1 instance. For example, to migrate all 7.1 instances to 8.1 on the same machine:

```
/usr/sbin/migrate-ds-admin.pl --oldsroot /opt/redhat-ds Gener\
al.ConfigDirectoryAdminPwd=password
```

Additionally migration scenarios are covered in the *Red Hat Directory Server Installation Guide*

4. Basic Information about Red Hat Directory Server

This is some basic information for using and managing Directory Server. The Directory Server information is explained in much more detail in the *Administrator's Guide*.

Starting and Stopping the Directory Server and Administration Server

The Directory Server and Administration Server instances are started and stopped using basic service command line tools. For example, on Red Hat Enterprise Linux:

```
service dirsrv-admin startservice dirsrv start
```

Running just **`service dirsrv start`** starts all instances of the Directory Server on the host machine. To start a sginle instance, use the name of the instance in the command:

```
service dirsrv start example
```

Starting the Directory Server Console

To start the Directory Server Console, run the **`redhat-idm-console`**.

```
redhat-idm-console
```

It is also possible to specify the user to log into the Console as using the `-u` and to give the URL to the Administration Server using the `-a` option.

```
redhat-idm-console -u "cn=Directory Manager" -a ht\
tp://ldap.example.com:9830
```

Default Port Numbers

These are the default port numbers for the Directory Server and Administration Server:

- The standard LDAP port is **`389`**.
- The secure (SSL) LDAPS port is **`636`**.
- The Administration Server port is **`9830`**.

Tool Locations

The Mozilla LDAP tools used to manage Directory Server, such as **`ldapsearch`** and **`ldapmodify`**, are in teh following directories, depending on platform:

- /usr/lib/mozldap6 on 32-bit Red Hat Enterprise Linux systems
- /usr/lib64/mozldap on 64-bit Red Hat Enterprise Linux systems
- /opt/dirsrv/bin/mozldap/ on HP-UX systems

Some OpenLDAP tools are located in /usr/bin on Red Hat Enterprise Linux systems already; it is possible to manage Directory Server with these tools (always using `-x` to disable SASL by default) but this is not recommended.

Directory Server File Locations

Red Hat Directory Server 8.1 conforms to the Filesystem Hierarchy Standards. For more information on FHS, see the FHS homepage, *http://www.pathname.com/fhs/*. The files and directories installed with Directory Server are listed in the tables below for each supported platform.