WEBADM INSTALLATION GUIDE

WebADM Installation Guide

1. Product documentation

This document is an installation guide for RCDevs WebADM Server. The reader should notice that this document is not a guide for installing WebADM applications (Web Services and WebApps). Specific application guides are available through the RCDevs Online documentation.

WebADM usage manual is not covered by this guide and is documented in the RCDevs WebADM Administrator Guide.

2. Product overview

WebADM is a powerful Web-based LDAP administration software designed for professionals to manage LDAP Organization resources such as Domain Users and Groups. It is the configuration interface and application server for RCDevs Web Services and WebApps such as OpenOTP or TiQR Server.

WebADM can be used standalone, as a powerful LDAP management console. It provides a hierarchical view of LDAP Organizations and many features for managing LDAP users and resources. It includes delegated administration (administrators can be created at different levels of the tree structure, with different privileges and views), supports multiple LDAP server at the same time, Domains, allows multiple authentication modes, provides comprehensive SQL and file-based audit trails, etc…

WebADM is compatible with Novell eDirectory, Microsoft ActiveDirectory 2008 & 2012, OpenLDAP, Apple OpenDirectory, Oracle/Sun Directory and RCDevs Directory Server. Other directories might work but are not tested nor supported by RCDevs. Other directories might work but are not tested or officially supported by RCDevs.

3. System requirements

The current version of WebADM runs on any Linux 32bit or 64bit operating systems with GLIBC >=2.5 and installed 32bit binaries. The installation package contains the required dependencies allowing WebADM to run on any Linux-based system without other requirement. WebADM only needs an LDAP backend (Novell eDirectory, OpenLDAP, RCDevs Directory Server or Microsoft ActiveDirectory) and a SQL database (MySQL, PostgreSQL, Oracle, SQLite). Other LDAP and SQL backends might work but are not officially supported.

For running WebADM and its applications, as well as the Radius Bridge Server and RCDevs Directory Server, your system should have the following requirements:

- A dedicated server computer or Virtual machine with Linux GLIBC >= 2.5 (RedHat, Centos, SuSe, Debian, Ubuntu).
- 1 GHz processor (multi-core / multi-thread processor is highly recommended). Both 32 and 64 bit chips are supported provided that 32 libraries are present.
- 2GB RAM memory
- 200-300MB disk space for installation files.
- Network access with DNS and a working NTP integration.
- A local or remote LDAP directory server (RCDevs Directory Server, OpenLDAP, Novell eDirectory or Microsoft ActiveDirectory >= 2008).
- A local or remote SQL database server (Ex. MySQL, PostgreSQL, Oracle, SQLite).
- Outbound Internet access for checking versions, connecting SMS gateways and sending emails.
- A local Mail Transfer Agent (Ex. Sendmail or Postfix).
- Firewall open ports: 80, 443, 8080, 8443, 1812.

4. Preliminary Information

WebADM relies on LDAP and most of the rest of this document is related to LDAP configurations. You should also be familiar with LDAP servers or know the basics of LDAP/AD administration in order to setup WebADM correctly.

Unlike other software, there is no “admin account” to be created in WebADM. Instead, you will login with your LDAP administrator account in the WebADM Administrator interface. The WebADM administrator account (referred as Super Admin below) is also generally your existing LDAP server’s administrator account. So the only accounts (admin or user) with WebADM are LDAP accounts.

The configurations described below talk about the WebADM LDAP proxy user and WebADM Administrator accounts. When you log in WebADM, you use an LDAP administrator account. The LDAP permissions and views inside WebADM also correspond to the LDAP permissions (ACLs) as configured and enforced by your LDAP server. This is also an LDAP configuration and not a WebADM configuration.

The WebADM proxy user is a special LDAP account which is used by WebADM to connect the LDAP server by himself (out of an admin session). For example, OpenOTP Server needs to search users and read/right user metadata in the LDAP. The proxy user is used by WebADM for such operations and also need sufficient LDAP permissions to handle theses tasks.

5. Installation and configuration

The WebADM installation is done in about 10 minutes and consists in 1) running the self-installer script, 2) configure the servers.xml and webadm.conf files, and 3) log in the WebADM admin console and run the graphical setup tasks.

The `/opt/webadm/conf/servers.xml` file contains the LDAP and SQL server configurations. It also contains Session Manager and PKI server configurations but you do not need to change these local services configurations in your base installation. The most important configuration here is your LDAP server connection.

The `webadm.conf` file contains the main WebADM configurations such as administrators, proxy user, LDAP containers used by WebADM to store its LDAP configurations, etc…

5.1. Installation types

RCDevs provides software packages for Linux as well as pre-installed VMWare Appliances. You can optionally get one of the VMWare Appliances as a base installation and modify its configuration instead of installing everything on a new server. Or you can use the Appliances directly for a quick start.

If you intend to deploy WebADM with Microsoft Active Directory and wish to use an Appliance as base installation, please use the OpenLDAP-based Appliance and drop the /opt/slapd directory to remove the pre-installed LDAP server. You will need to reconfigure the configuration files to setup the LDAP connection to your domain controller(s) and to adjust some configuration settings.

The VMWare Appliances already have WebADM, RadiusBridge, MySQL, LDAP configured and have all the WebADM applications pre-installed (including OpenOTP). Always consider doing your first tests using an unmodified Appliance (with the pre-installed LDAP), in order to accommodate with the software and its functionalities. Once you manage to use WebADM and play with the interface, it will be much easier to install and run you WebADM packages on one dedicated Linux server.

5.2. Install WebADM

The installation package is built for the i386/i686 Linux architecture. 64bits Linux systems also need to have the Glibc package for i686 installed. Moreover the WebADM installer requires the ‘make’ command. Be sure that GNU Make is correctly installed.

On a RedHat, Centos or Fedora system, you can install the required i686 packages with:

```
yum install glibc.i686 libgcc.i686
yum install make
```

On a Debian or Ubuntu, you can install the required i686 packages with:

```
apt-get install libc6-i686 ia32-libs
apt-get install make
```

> **Note**
>
> If you install WebADM in high availability mode (on several servers), please refer to the WebADM High Availability Guide for the cluster setup instructions.

5.2.1 Install with yum repository

On a RedHat, Centos or Fedora system, you can use our repository, which simplify updates.

Add the repository:

```
curl http://www.rcdevs.com/repos/redhat/rcdevs.repo -o /etc/yum.repos.d/rcdevs.repo
```

Clean yum cache and install WebADM with all WebApps & Services:

```
yum clean all
yum install openid openotp opensso pwreset selfdesk selfreg smshub spankey tiqr webadm
```

Run the setup script:

```
/opt/webadm/bin/setup
```

It initializes the WebADM PKI, etc…

5.2.2 Install with Debian repository

On a Debian system, you can use our repository, which simplify updates.

Add the repository:

```
echo "deb http://rcdevs.com/repos/debian ./" > /etc/apt/sources.list.d/rcdevs.list
apt-key adv --fetch-key http://rcdevs.com/repos/debian/RPM-GPG-KEY-rcdevs.pub
```

Clean cache and install WebADM with all WebApps & Services:

```
apt-get update
apt-get install openid openotp opensso pwreset selfdesk selfreg smshub spankey tiqr webadm
```

Run the setup script:

```
/opt/webadm/bin/setup
```

It initializes the WebADM PKI, etc…

5.2.3 Install using the self-installer

You first need to download and install the WebADM software package. You can download the latest package on the RCDevs Website. Download and copy the WebADM-all-in-one self-installer package to your server. You can copy the package file to the server with WinSCP or SCP. Then connect via SSH to your server, uncompress and run the self-installer package with:

```
gunzip webadm-all-in-one-1.x.x.sh.gz
bash webadm-all-in-one-1.x.x.sh
```

The installation process will automatically run the console-based setup script in bin/setup. This setup script creates the WebADM system user, filesystem permissions, initializes the WebADM PKI, etc…

5.3. Setup the SQL database

WebADM uses a database to store audit logs and localized messages. Application configurations, users and their metadata are directly stored in LDAP rather than in the databases. WebADM supports both MySQL and PostgreSQL databases. Other databases are not currently fully tested. You must create a webadm database on your SQL server and a webadm user with password webadm, having full permissions on that database. Edit the `conf/servers.xml` file and adjust the SQL Server parameters such as the database user name and password.

5.4. Setup the LDAP directory

WebADM relies on one or several LDAP directories for storing and managing user resources, groups, policies and applications configurations. It supports RCDevs Directory Server, Novell eDirectory, OpenLDAP, 389 and Microsoft ActiveDirectory.

⚠ Important

ActiveDirectory is supported on Windows Server higher or equal to 2008 R2. Please do not install WebADM with Windows servers prior to version 2008 like Windows Server 2003!

5.4.1. With RCDevs Directory Server

You can find the installation documentation here.

Edit the `conf/servers.xml` file and adjust the LDAP Server parameters. If WebADM and RCDevs Directory Server (DS) are installed on the same server, it is recommended to use no encryption (*encryption="NONE"*) and port 389. If WebADM and DS are running on different servers, it is recommend to use TLS encryption (*encryption="TLS"*) on port 389.

The RCDevs Directory Server is pre-populated with a base structure and the admin DN is `cn=admin,o=root`. This admin DN is required to login WebADM for the first time.

5.4.2. With Novell eDirectory & Oracle Directory

Edit the `conf/servers.xml` file and adjust the LDAP Server parameters. It is recommended to use TLS encryption with Novell eDirectory on port 389 as the default eDirectory installation does not allow unencrypted connections. Please refer to your eDirectory documentation to use eDirectory without TLS/SSL.

5.4.3. With OpenLDAP

Edit the conf/servers.xml file and adjust the LDAP Server parameters. It is recommended to use no encryption (encryption="NONE") with OpenLDAP and port 389. If your OpenLDAP server supports SSL/TLS, you can alternatively use SSL encryption on port 636 or TLS on port 389.

You need to register the WebADM OpenLDAP schema in your OpenLDAP server. If your OpenLDAP uses a flat file for configuration (Ex. `/etc/openldap/slapd.conf`) then proceed this way:

- Copy the `/opt/webadm/doc/OpenLDAP.schema` to `/etc/openldap/schema/webadm.schema`
- Edit the `/etc/openldap/slapd.conf` file and add the following line:
  `include /etc/openldap/schema/webadm.schema`
- Add admin ACLs for the WebADM proxy user in the `slapd.conf` file:

```
Example:
 access to *
 by dn="cn=webadm,dc=WebADM" write
  …
```

Replace cn=webadm,dc=WebADM with your WebADM proxy user DN (see section 4.5 for details). For an easier setup, you can set the proxy user to the LDAP default admin user and password.

- Restart your OpenLDAP server.

If your OpenLDAP uses the newer cn=config as the main configuration instead of slapd.conf then adding additional schemas can be done graphically in WebADM. Follow the rest of the instructions and at first login, you will be prompted for some graphical setup tasks including the schema additions.

5.4.4. With Microsoft Windows Server 2008 Active Directory

You must enable LDAP SSL for Active Directory to use WebADM. LDAP over SSL is a requirement in Active Directory for managing user passwords and create Active Directory users from the Admin Portal. If the Windows Server Certificate Authority (CA) is not installed, install it first on your Active Directory server as follows, in order to activate the ActiveDirectory LDAP v3 protocol with SSL:

- Click `Start` => `Administrative Tools` => `Server Manager`.
- In the Roles Summary section, click `Add roles`.
- On the Select Server Roles page, select the `ActiveDirectory Certificate Services` check box. Click `Next` two times.
- On the Select Role Services page, select the Certification Authority check box, and then click `Next`.
- Follow the procedure provided to setup an Enterprise CA.

Note

With Active Directory, you need install the certificate authority service components to activate Active Directory over SSL.

> **Note**
>
> Windows server need to be restart for the SSL to be activated.

Edit the WebADM servers configuration file in conf/servers.xml and change the LDAP port to 636 and encryption to SSL (encryption="SSL").`

You have two way to setup WebADM LDAP schema for Active Directory:

1. With the WebADM schema extension (preferred).
2. Without any schema addition (re-uses existing object classes and attributes as a replacement).

5.4.4.1. With WebADM Schema Extension

This option is preferred and WebADM will use the RCDevs IANA-registered Active Directory attributes to store additional LDAP data in users and groups. The WebADM schema addition is very minimal and is composed of 3 new object classes (webadmAccount, webadmGroup and webadmConfig) and 3 new attributes (webadmSettings, webadmData and webadmType).

If you choose this installation option, then you must connect WebADM to the domain controller having the Schema Master Role in Active Directory to let WebADM register its schema additions. If you connect WebADM to two domain controllers in the servers.xml file, the first one one should be the one with the Schema Master Role. Without it, the WebADM graphical setup (explained later) will not be allowed to add the required object classes to your Active Directory.

5.4.4.2. Without WebADM Schema Additions

With this option, WebADM does not make any addition to the Active Directory schema. Instead the configuration WebADM is customised to re-use some existing object classes and attributes. Please go to directory `doc/ActiveDirectory/Schema_Not_Extended/` and copy the files `webadm.conf` and `objects.xml` to the WebADM directory `conf/`. The following changes are applied to the configurations:

In `conf/webadm.conf`, the default configurations:

```
webadm_account_oclasses "webadmAccount"
webadm_group_oclasses "webadmGroup"
webadm_config_oclasses "webadmConfig"
webadm_data_attrs "webadmData"
webadm_settings_attrs "webadmSettings"
webadm_type_attrs "webadmType"
```

are changed to:

```
webadm_account_oclasses "bootabledevice"
webadm_group_oclasses "bootabledevice"
webadm_config_oclasses "device"
webadm_data_attrs "bootFile"
webadm_settings_attrs "bootParameter"
webadm_type_attrs “serialNumber"
```

WebADM will also use the AD object class bootabledevice as user/group activation class and the object class device for the LDAP configuration objects' storage. It will also store user settings and metadata in the bootFile and bootParameter attributes in the class bootabledevice.

In `conf/objects.xml`, the LDAP object specifications are configured to use the replacement object classes and attributes.

5.5. Edit the WebADM main configuration file

The LDAP and SQL connections are now prepared and it is now time to customize the WebADM server settings in `/opt/webadm/conf/webadm.conf`. Please read the configuration file even if you do not need to change the settings as all the main configurations are documented inline.

Note

With RCDevs Directory Server, the default configuration does not need to be modified and you can skip this section.

The proxy user, admin users and containers in LDAP are specified with LDAP DN. Please look at your LDAP documentation for understanding DN syntaxes. With OpenLDAP, and Active Directory, you must know your LDAP tree base DN before starting configuring WebADM. Generally this is something like o=YourCompany or dc=YourDomain. With Active Directory, The base DN is like `dc=YourDomain,dc=com`.

Most of the WebADM settings are pre-configured to work out-of-the-box with any of the supported LDAP backends with very minor changes. Just be sure to change the LDAP tree base (suffix) in the following settings according to your LDAP server:

- proxy_user
- super_admins
- other_admins
- optionsets_container
- webapps_container
- websrvs_container
- mountpoints_container
- domains_container
- clients_container

The correct DN for the above settings may vary depending on your LDAP server type and LDAP tree base DN. For example, if your LDAP tree base is an Organization object, (ex. o=Mydomain), then the proxy user DN should be cn=webadm,ou=WebADM,o=Mydomain. And all the subcontainers should use OU objects (ex. ou=OptionSets,ou=WebADM,o=Mydomain).

Check your LDAP server documentation for the setting up WebADM with the correct DN syntaxes. If you use Active Directory and are not aware of these DN syntaxes or do not know what is your AD tree base, then you should first install an LDAP / AD explorer tool for Windows. You will also be able to browse your directory structure , check what is your AD tree base and what are the user DN for the admin / proxy users.

Example

Change optionsets_container “dc=OptionSets,dc=WebADM” To “ou=OptionSets,ou=WebADM,o=MyDomain”

The proxy user is required by WebADM to access LDAP resources (ex. configuration and users) out the permissions of the working users. The proxy user must have at least read-only permissions on the whole LDAP tree. If you are using WebApps and Web Services such as OpenOTP, the proxy user requires administrator permissions on the trees where are stored the LDAP users. With Novell eDirectory, the WebADM graphical setup will create the proxy user and its permissions for you.

With OpenLDAP, the proxy user can be created later by the graphical setup but you need to add admin permissions in the `/etc/slapd.conf` file (see in section 4.4.3).

With Microsoft ActiveDirectory, you must use an existing Domain user for the proxy user and it must be part of the Domain Admins group. You can use your Administrator account as proxy user. In this case set `CN=Administrator,CN=Users,DC=MyDomain,DC=COM` as proxy user and replace the tree base `DC=MyDomain,DC=COM` with your own tree base.

The WebADM super administrators (defined in the main WebADM configuration file) have unrestricted access to WebADM resources and rights to run the WebADM graphical setup. You must set the super administrator accounts to one or more existing LDAP users (using DN syntaxes) and use one of these accounts to enter the WebADM Admin Portal.

The `conf/object.xml` file contains customizations and display handler assignments for LDAP objectclasses and attributes. Do not modify this file unless you really know what you are doing.

WebADM configurations are cached in memory. If you change configurations, please restart WebADM or purge the config cache in the Infos menu in the WebADM Admin Portal.

5.6. Set the listener ports

You can edit the bin/webadm startup script if you need to change some runtime configurations such as allocated memory and default HTTPd / SOAPd listener ports. By default the Web server listens on port 443 (SSL) and the SOAP server listens on ports 8080 and 8443 (SSL). It is recommended to keep the default port settings. If you need to change a port for example, do not modify the `bin/webadm` file and prefer creating a `conf/webadm.env` file where you re-define the configuration variables you need to change in this file. This is preferred because the next upgrade will override the `bin/webadm` file. Look at the `bin/webadm` for variable syntax.

5.7. Start WebADM and run the graphical setup

Start WebADM with the command `bin/webadm start`.

Enter WebADM with your super administrator account and run the graphical setup. The login URL is `https://<your-server-address>`. Only a super user can run the graphical setup.

Important

Until the graphical setup is done and at least the first WebADM Domain is created, you must login with the administrator LDAP DN and not the username (DN login mode). The administrator LDAP DN is the DN you have defined for the super_admins setting in webadm.conf. When the graphical setup will be completed and at least one Domain is created for the LDAP tree where you administrator is stored, you will be able to login with username and password (UID login mode)

Important

If you use RCDevs Directory Server, the admin DN is *cn=admin,o=root*.

The Setup button will appear in the home page when you enter the WebADM Admin Portal.

WebADM requires DN-based login until the setup is completed. Then it will use the login mode as configured in the `conf/webadm.conf` file.

WebADM will run very slow and will not be functional until the graphical setup has been completed. The MountPoints, OptionSets, WebSrvs, WebApps and many features are kept disabled until the setup is completed.

The graphical setup process will:

- Create the required database tables (as specified in the `conf/database.xml` file).
- Register the required LDAP schema objectclasses and attributes (with Novell eDirectory and Microsoft ActiveDirectory).
- Create the proxy user (if not already existing).
- Setup the proxy user permissions (on Novell eDirectory).
- Create the WebADM LDAP containers (as defined in the `conf/webadm.conf` file).

If WebADM fails to automatically create the LDAP containers, create the containers manually with the object creation wizards.

5.8. Configure your authentication method

By default, WebADM is configured with DN login mode (auth_mode) meaning you log in with the user's LDAP DN and password. Working in this mode is required until the WebADM setup is completed.

Note

If you use a VMWare Appliance, it is configured with UID login mode. Please change to DN login mode if you connect the Appliance to another LDAP server.

When the graphical setup is completed, you can switch the login mode to UID or PKI. Please note that PKI mode required that you create a login certificate in WebADM for your administrator account. Be sure to restart WebADM when you update the configuration files.

- PKI login mode uses user certificate and LDAP password and is the most secure and recommended login method. To use PKI authentication, you must first log in WebADM, create an Admin certificate for your administrator user, and install it in your Web browser.
- UID login mode uses a username, domain and LDAP password. A WebADM Domain must have been created for the LDAP context where the administrator account is stored (See the WebADM Administrator Guide and look for WebADM Domains for details). Note that the WebADM graphical setup should have already created a first domain (called Default) for the LDAP context of your administrator account.

Please read the `conf/webadm.conf` file comments for more explanations about the login modes. When a domain exists, you can configure WebADM with UID login mode. In this mode, you log in with a username and password (not with an LDAP DN). But until setup is completed, WebADM always enforces DN login mode even if UID mode is configured.

5.9. Check system clock and timezone

WebADM requires an accurate system clock and timezone. Your Linux server should be configured with NTP time synchronization. On RedHat/CentOS, you need to install and run the ntpd service at boot time. After installing ntpd, you can check the server time with the ntpdate command.

WebADM before version 1.5.6 required the time zone to be configured in webadm.conf. With later versions, WebADM uses the time zone which is configured at the system level. On most Linux systems, the timezone is configured by adjusting the `/etc/localtime` file or symlink.

5.10. Check the logs

During installation look at the SQL logs (in the WebADM Database menu) and at the log files in the `/opt/webadm/logs/` folder to track configuration and runtime errors. There will be many errors until the graphical setup is completed. This is a normal behavior.

The httpd.log contains all the log events related to the WebADM administration and the user operations in the WebApps (ex. Self-Service). And the soapd.log contains all the log events related to service operations (ex. OpenOTP log events).

6. Upgrades

If you use our rpm or deb repository, WebADM is upgraded like other packages with `yum update` or `apt-get upgrade`.

If you don't use it, you need to download and install the latest version. When upgrading WebADM, do not remove the previous version and proceed exactly like for installations by running the selfinstaller.

Upgrade will not override your configuration files and will update the .default configuration files. Check the content of the .default files for changes and modify your configuration files accordingly.

After an upgrade please read the RELEASE_NOTES, CHANGELOG and README files to get the list of changes and follow the recommendations if any.

⚠ Important

After an upgrade, the WebApps and Web Services configurations may need to be updated. Log in the WebADM Admin Portal and check the installed applications status in the home page. If a configuration update is required, click the Not Configured link (in red), check the settings and save the application configuration. Your application status should be valid again.

7. Usage

You can start WebADM with the webadm controller script in `/opt/webadm/bin/`. Once started and installed, administrators can enter WebADM Administrator Portal under the following URL: `https://SERVER_ADDR/`.

End-user can administrate they own account data from the User Self Service Desk (SelfDesk) WebApp. The WebADM WebApps portal URL is `https://SERVER_ADDR/webapps/`.

⚠ Important

To be able to use any WebADM application (WebApp or Web Service), an LDAP user must be a WebADM-enabled account. This means usable LDAP accounts are those containing the webadmAccount LDAP objectclass. In WebADM, administrators can enable the WebADM features on any LDAP user / group by extending it with the webadmAccount objectclass (with the *Add Extension* user action).

RCDevs solutions (OpenOTP, TiQR, OpenID, etc…) run on top of the WebADM Server. The solutions are generally composed of both Web Services and end-user Web Applications. WebADM is an application server which provides the HTTP and SOAP engines required by Web Services and WebApps.

7.1. WebADM End-User Applications (WebApps)

A WebADM Web Application is a pluggable component to be installed (deployed) in WebADM. WebApps are generally companion application for Web Services. For example, OpenOTP Software Token requires the end users to register their secret token keys, resynchronize their token application, etc… One other example is the OpenID authentication and redirection page.

The Web Applications provide:

- Some public web pages
- Optional authentication with PKI, or Domain Login (depending on the WebApp purpose)
- A graphical configuration in the WebADM Applications menu

Note

Each WebApp is accessible through a specific URL in WebADM. This URL provides access to all the resources (stylesheets, images…) required by the WebApps to run without accessing any other WebADM Web location. This allows network administrators to restrict the Administrator Portal for internal use only, and publish the WebApps specific URLs on the internet using Apache reverse proxies.

7.2. WebADM Web Services

A WebADM Web Service is a pluggable component to be installed (deployed) in WebADM. The Web Services provide final functionalities such as user authentication services.

The Web Services provide:

- A SOAP XML interface
- A WSDL service description file
- A graphical configuration in the WebADM Applications menu

Note

WebADM supports only HTTP-based Web Services (i.e. SOAP/XML). Other APIs require external components such as the OpenOTP RADIUS Bridge to be installed. Radius Bridge enables the RADIUS service API for OpenOTP.

8. Tuning LDAP Permissions

8.1. WebADM Proxy User

There are two things to be considered in order to implement fine-grained LDAP permission for WebADM and its applications.

1. WebADM Proxy user permissions: This system user is used by WebADM to access and manipulate the required LDAP resources.
2. Administrator users permissions: These accounts login to the Admin portal in order to manage LDAP resources and registered applications.

The proxy user is required by WebADM to access LDAP resources (ex. application configuration, users, groups…) out the permissions of an Admin user's session.

The proxy user must have at least read-only permissions on the whole LDAP tree. It is used by the WebApps and Web Services such as OpenOTP and also requires some attribute write permissions as described below, over the trees where are stored the LDAP users. By default and for simplification, it is recommended to use an Administrator account of the LDAP directory as WebADM Proxy user.

If you need to implement finer LDAP access rights then:

1. Proxy user needs to perform wide LDAP search and reads. It also requires read-only permissions to the WebADM LDAP configurations (ie. configured containers) and to the user Domains subtrees.
2. Proxy user needs to do some write operations to a few LDAP attributes because it needs to store dynamic application user data into the users.

In some circumstances, the Proxy user will also need to write application setting on the users and groups. The following attributes are part of the WebADM LDAP schema and need Proxy user write permissions:

- webadmData : is the attribute where the applications store the user data (ex. OpenOTP enrolled Token states).
- webadmSettings : is the attribute where WebADM stores user-specific settings (ex per-user OTP policy).

If you use WebADM Self-Services, and depending on what you allow users to do within the selfservice applications, then WebADM Proxy user may need some additional permissions: Ex. if you want users to reset their LDAP password, set their mobile numbers or email addresses, then the Proxy user will need to have write permissions to the corresponding LDAP attributes.

In general, it is recommended to implement Proxy user write access to the following attributes:

- webadmData (dynamic and encrypted application data)
- webadmSettings (only if Self-Services are used to configure account settings)
- mail (only if Self-Services are used to set email addresses)
- mobile (only if Self-Services are used to set mobile numbers)
- preferredLanguage (only if Self-Services are used to set user language)
- userPassword or unicodePwd for Windows AD (only if Self-Services are used to set user password)

8.2. Administrators

When an administrator logs in the WebADM Admin Portal, he always accesses and manages the LDAP resources under his own LDAP permissions. This means the user/group/configuration management permissions are enforced at the LDAP level. For example, a Windows AD Domain Administrator will be able to manage users and groups.

Note

To be able to log in WebADM, an LDAP user must be part of either WebADM super_admins or other_admins. You can also create read-only Administrators by adding standard users to the other_admin group.

9. Active Directory Notes

Please read section 5.4.4 to configure WebADM for Active Directory. This is a summary of the changes to be made to your WebADM configurations for use with Microsoft Active Directory.

In a first step you need to configure your AD Domain Controllers with LDAP SSL and you need to know what is the exact LDAP base DN for your AD. You can use an LDAP explorer tool for windows in order to know what is your LDAP base DN. It should look like `dc=yourDomain,dc=com`.

Then change the following WebADM configurations :

1) Configure the `conf/servers.xml` file.

Change the hostname (i.e. the host setting) to your AD Domain Controller IP or hostname. Please connect first the Domain Controller having the Schema Master Role in your AD (at least until you completed the whole setup).

You configuration should look like:

```
<LdapServer name="My AD Server"
 host="MyDCServerIP"
 port="636"
 encryption="SSL"
 cert_file=""
 key_file="" />
```

Please notice the port 636 for LDAP SSL on Active Directory.

2) Configure the `conf/webadm.conf` file.

You need to adjust all the settings containing an LDAP DN.

- proxy_user: Use your usual AD Administrator DN as proxy user. It should be something like cn=Administrator,cn=Users,dc=yourDomain,dc=com. You can use another user here but the proxy user must have Admin rights to the other LDAP user objects.
- proxy_password: This is the AD password for the proxy user (i.e. your Administrator password).
- super_admins: Put again the same Administrator DN in this administrator user/group list.
- other_admins: You can comment it for your initial setup.
- optionsets_container and all the other containers: You need to adjust all these settings with your AD LDAP base DN (i.e the AD LDAP suffix). For example, optionsets_container should look like `cn=OptionSets,cn=WebADM,dc=yourDomain,dc=com` with `dc=yourDomain,dc=com` as example LDAP base DN.
- alert_email: Put your email address here or the distribution list of your system administrators.

3) Restart WebADM and login at `https://yourserver/`.

WebADM will ask for your Administrator LDAP DN. In our example enter `cn=Administrator,cn=Users,dc=yourDomain,dc=com` and the corresponding password.

Please look at WebADM admin logs by connecting the WebADM server with SSH and running the command `tail -f /opt/webadm/logs/httpd.log` to track down login issues.

4) After login, WebADM will prompt you for running some required graphical setup tasks. Just follow the instructions.

5) Configure your first WebADM Domain. Go to `WebADM menu` -> `Infos` -> `Registered Domains`. Adjust the default Domain object if necessary or create a new Domain if none exist. The Domain User Search Base setting must be set to the LDAP location (i.e. the container node) where you have your users (example: cn=Users,dc=yourDomain,dc=com).

6) Configure your applications in `WebADM menu` -> `Applications`. At least, you should set the Default Domain setting to your WebADM Domain for all the registered applications.

7) Until now your WebADM server was configured with DN login mode to the Admin portal.

This means you had to login with the Administrator full DN and password. But now that a WebADM Domain is created for locating the users inside the AD LDAP, you can re-edit the `conf/webadm.conf` file and change the auth_mode setting to UID. With UID mode, you can login to WebADM with the AD Administrator username instead of the full DN. But this works only when the WebADM Domain is defined correctly. You can also switch back to the DN mode at any time in case of a problem.

10. OpenLDAP Notes

Unlike other directories, OpenLDAP user password values can be read from the userPassword attributes through the LDAP API. Your OpenLDAP server must also be configured to use password hashing in order to store non-reversible password hashes in the user objects. Hashed passwords are completely secure and the user passwords can also not be recovered from the hashed values.

RCDevs Directory Server is configured with automatically hashing of cleartext passwords in the userPassword attributes. User password changes performed from UNIX command line, OpenLDAP tools, WebADM Self-Services or WebADM Admin Portal will also be hashed with SSHA (Salted Secure Hash Algorithm) prior to be stored in user objects.

Other OpenLDAP implementations (used in common Linux distributions) may not use cleartext password hashing by default. You should enable the feature by editing your OpenLDAP slapd.conf configuration file and add the following configurations.

```
overlay ppolicy
ppolicy_hash_cleartext
password-hash {SSHA}
```

The recommended hashing method for LDAP servers is SSHA. Yet SHA or MD5 can be used too. Please read your OpenLDAP documentation for details.

You can check whether your OpenLDAP is configured with password hashing by setting a user password and exporting the user object to LDIF from the WebADM user edit. If the user password appears in clear in the LDIF export, then password hashing is not enabled. And if you see a password value like *{SSHA}6C5wmU7E8uGwuGF+bUDtI1ivRNlOdJJu*, then password hashing is enabled with SSHA.

If you prefer keeping your OpenLDAP configurations unmodified and the OpenLDAP server does not hash LDAP passwords, you can configure WebADM to enforce the password hashing function on behalf of the LDAP server. To enable WebADM password hashing on password attribute creation or change, open the configuration file `/opt/webadm/conf/objects.xml` and change the block:

```
<Attribute name="userpassword"
 desc="Password"
 handler="password.php"
 advanced="yes" />
```

to

```
<Attribute name="userpassword"
 desc="Password"
 handler="password.php"
 advanced="yes"
 encoding="ssha" />
```

Do not enable WebADM password encoding with SSHA with other LDAP implementations than OpenLDAP.