



WEBADM UPGRADE GUIDE FROM 1.3 AND OLDER TO 1.4 AND LATER

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs.

Copyright (c) 2010-2017 RCDevs SA. All rights reserved.

<http://www.rcdevs.com>

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to info@rcdevs.com.

WebADM Upgrade Guide from 1.3 and older to 1.4 and later

[upgrade](#)

1. Introduction

This document provides the necessary information for upgrading servers running WebADM v1.3 to WebADM v1.4 released in July 2015. WebADM v1.4 is a major upgrade of RCDEVs WebADM which includes major changes listed at the end of this document. The reader should notice that this document is not a guide for installing WebADM or its 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 available through the RCDevs' online documentation website. In this guide, we will consider a common installation scenario where WebADM is running in cluster mode with two active servers.

IMPORTANT

WebADM v1.4 is NOT compatible anymore with Microsoft Active Directory 2003. You need at least Microsoft AD 2008-R2 for WebADM v1.4.

2. Backup Your Server

Before upgrading, please back up your WebADM configurations on any WebADM server:

- > Connect the servers via SSH or Putty.
- > Backup the configuration folder with:

```
tar -zcvf /root/webadm_conf.tgz /opt/webadm/conf
```

- > Backup the PKI folder with:

```
tar -zcvf /root/webadm_pki.tgz /opt/webadm/pki
```

3. Install the Upgrade Package

You will need to perform the following instructions on both servers of your cluster.

- > Download the WebADM v1.4 all-in-one package.
- > Copy the package on the servers with WinSCP or another SCP tool.
- > Connect the servers via SSH or Putty.
- > Uncompress the package with:

```
gunzip webadm_all_in_one-1.4.x64.sh.gz
```

› Run the upgrade tool with the commands:

```
chmod 755 webadm_all_in_one-1.4.x64.sh; ./webadm_all_in_one-1.4.x64.sh
```

› Choose upgrade (choice U) and choose not to restart WebADM. Note: Please uncompress the GZ archive once copied on the server and not under Windows!

4. Adjust the Server Configurations

You will need to a) adjust some OS-level configurations on the servers and b) edit the WebADM configuration files to make the necessary changes for WebADM v1.4.

4.1 OS-Level Configurations

If you run a cluster, you need to configure one additional port for the communication between your cluster node. In WebADM v1.3, the session server and the session replication used TCP and UDP port 11211. In WebADM v1.4, the session server and the session replication uses one unique port: TCP 4000.

Edit the firewall configuration in `/etc/sysconf/iptables`. Comment the following two lines if present:

```
-A INPUT -p tcp -m tcp -m multiport -s <secondary node IP> -j ACCEPT --dports 11211,11212  
-A INPUT -p udp -m udp -m multiport -s <secondary node IP> -j ACCEPT --dports 11211,11212
```

Now add the following line after the two commented lines:

```
-A INPUT -p tcp -m tcp -s <secondary node IP> -j ACCEPT --dport 4000
```

4.2 WebADM Configurations

You will need to make changes on 2 WebADM configuration files: `webadm.conf` and `servers.xml`.

4.2.1 `webadm.conf`

Edit `webadm.conf` and locate the setting `optionsets_containers`. Add the configuration line for `adminroles_containers`. With OpenLDAP or Novell your configuration should look like:

```
# WebADM Optionsets container
optionsets_container "dc=OptionSets,dc=WebADM"
# WebApp configurations container
webapps_container "dc=WebApps,dc=WebADM"
# WebSrv configurations container
websrvs_container "dc=WebSrvs,dc=WebADM"
# Mount points container
mountpoints_container "dc=MountPoints,dc=WebADM"
# Domain and Trusts container
domains_container "dc=Domains,dc=WebADM"
# Clients container
clients_container "dc=Clients,dc=WebADM"
```

After the change it should look like:

```
# WebADM AdminRoles container
adminroles_container "dc=AdminRoles,dc=WebADM"
# WebADM Optionsets container
optionsets_container "dc=OptionSets,dc=WebADM"
# WebApp configurations container
webapps_container "dc=WebApps,dc=WebADM"
# WebSrv configurations container
websrvs_container "dc=WebSrvs,dc=WebADM"
# Mount points container
mountpoints_container "dc=MountPoints,dc=WebADM"
# Domain and Trusts container
domains_container "dc=Domains,dc=WebADM"
# Clients container
clients_container "dc=Clients,dc=WebADM"
```

If you use Microsoft Active Directory, the configuration should look like below with a domain name specific to your organization:

```
optionsets_container "cn=OptionSets,cn=WebADM,dc=mydomain,dc=com"
webapps_container "cn=WebApps,cn=WebADM,dc=mydomain,dc=com"
websrvs_container "cn=WebSrvs,cn=WebADM,dc=mydomain,dc=com"
mountpoints_container "cn=Mountpoints,cn=WebADM,dc=mydomain,dc=com"
domains_container "cn=Domains,cn=WebADM,dc=mydomain,dc=com"
clients_container "cn=Clients,cn=WebADM,dc=mydomain,dc=com"
```

After the change it should look like:

```
adminroles_container "cn=AdminRoles,cn=WebADM,dc=mydomain,dc=com"
optionsets_container "cn=OptionSets,cn=WebADM,dc=mydomain,dc=com"
webapps_container "cn=WebApps,cn=WebADM,dc=mydomain,dc=com"
websrvs_container "cn=WebSrvs,cn=WebADM,dc=mydomain,dc=com"
mountpoints_container "cn=Mountpoints,cn=WebADM,dc=mydomain,dc=com"
domains_container "cn=Domains,cn=WebADM,dc=mydomain,dc=com"
clients_container "cn=Clients,cn=WebADM,dc=mydomain,dc=com"
```

4.2.2 server.xml

The session server port change has to be configured in servers.xml. Edit the servers.xml and locate the SessionServer blocks. Change all the occurrences of port="11211" to port="4000".

Before the change your configuration should look like:

```
<SessionServer name="Session Server 1"
host="primary node IP"
port="11211" />
<SessionServer name="Session Server 2"
host="localhost"
port="11211" />
```

After the change it should look like:

```
<SessionServer name="Session Server 1"
host="primary node IP"
port="4000" />
<SessionServer name="Session Server 2"
host="localhost"
port="4000" />
```

4.3 Restart Servers

At this stage, you can restart your servers one by one. If you didn't configure WebADM 1.4 correctly, the server will complain and tell what configuration directive is wrongly configured.

5. Adjust Application Configurations

Login on Admin Portal on any of your WebADM servers. On the home screen, WebADM will display a message telling that the installation is incomplete.

Click the Run Setup Wizard link. At the bottom of the screen you should see the following messages:

»

Click the Create default containers and objects button.

»

WebADM has created your AdminRoles container. If you had other_admins configured in your webadm.conf file, then it has also created a default role and has configured your other_admins users or groups as members to the AdminRole. In the left tree you should see the new container:

»

IMPORTANT

WebADM 1.4 does not require the other_admins configuration anymore. Instead, it uses AdminRole objects. Please read the WebADM Manual for details on AdminRoles.

6. Adjust Application Configurations

WebADM 1.4 comes with upgrades for all your registered applications. Some of them may be missing some mandatory settings after the upgrade. In this case, go to the Applications menu and click the CONFIGURE button for any application where the Not Configured message is displayed in red.

»

Scroll down your application configurations and simply click the Apply button. WebADM will apply the default configuration for any new setting.

7. MySQL & MariaDB Issues

With MySQL and MariaDB you might get connection errors with “Host ‘hostname’ is blocked because of many connection errors; unblock with ‘mysqladmin flush-hosts’”. This is due to the socket polling system of WebADM Watchd which checks the database connection is available by opening TCP sockets every 10 seconds. Recent versions of MySQL count these socket checks as failed connections and also blacklists the WebADM servers after 100 checks. The workaround simply consists of re-configuring the failed connections’ limit to the maximum value in MySQL.

On 32bit MySQL servers, set max_connect_errors=4294967295 On 64bit MySQL servers, set max_connect_errors=18446744073709551615

You should now be able to work with WebADM v1.4 or later!

This manual was prepared with great care. However, RCDevs S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language without the prior written permission of RCDevs S.A. The latter especially applies for data processing systems. RCDevs S.A. also reserves all communication rights (lectures, radio and

television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs S.A. © 2021 RCDevs SA, All Rights Reserved