

Oct 21, 2020 02:44

## How to configure Unified CDR-View in WMS Network

**i** This document explains how to configure a system to have a unified CDR-View in a WMS Network.

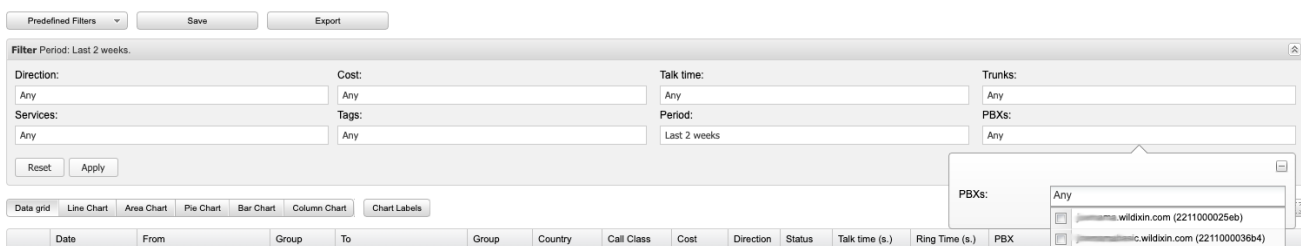
Created: September 2020

Permalink: <https://confluence.wildix.com/x/GwKmB>

- [Introduction](#)
- [Requirements](#)
- [Configuration](#)
  - [Server configuration](#)
  - [Client configuration](#)
  - [Result](#)
- [Final Considerations](#)

## Introduction

We know in Wildix it is very easy to create a [network of several systems](#). Once the network is created, it is completely transparent to end users whether they are in a system that is part of a network or not. The systems on the network appear as a single system to end users. When in a WMS Network environment, there is even a filter in CDR-View to filter reports based on which PBX the activity happened on. See the filter in the screenshot below.



However, there is some configuration needed in order to have the filter to work. Without such configuration the systems on the WMS Network can only access their own information. The purpose of this document is to explain the needed configurations to have a unified CDR-View in a WMS Network.

## Requirements

The only requirement to have a unified CDR-View is to have all of the systems configured to use the same database. Out of the box, every system uses a local database. So, first of all we need a database server that is accessible to all of the systems on the WMS network. For the sake of simplicity in this document we use one of the systems as a database server for the rest of systems.

**Note:** if you have an external database server you just have to recreate the database structure (use mysqldump), or even take a full dump of one of the servers database and apply it in your database server.

## Configuration

Here we take advantage of the fact that, once part of a WMS Network, systems are part of a secure intranetwork. In this scenario we have 2 systems:

IP	Serial	Host name	Failover
5.15.0.1	2211000025eb	<a href="#">jemma.wildix.com</a>	
5.15.0.3	2211000036b4	<a href="#">jemma.wildix.com</a>	

As you can see, systems communicate to each other using the displayed IP addresses. For this example, we use system 2211000025eb (IP 5.15.0.1) as the database server.

### Server configuration

The database server needs to be accessible by all of the other systems. Since here we are using one of the systems as the database server, we don't need to worry about the database structure, it is already there. We just need to make it available:

- On the 5.15.0.1 server, login to mysql console (Password is wil01dix):

```
$mysql -u root -p
Enter password:
```

- Execute the following command:

```
> show grants;
```

```

MariaDB [(none)]> show grants;
+-----+
| Grants for root@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY PASSWORD '*47C562035B611FE7E0CD4CF0F58AEC9EDEF20E3' WITH GRANT OPTION |
| GRANT PROXY ON ''@%' TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
2 rows in set (0.000 sec)

```

- As you can see in the previous picture, the login for the root account is only available from the localhost. We need to execute the following command in order to enable access from remote locations:

```
>GRANT ALL PRIVILEGES ON *.* TO 'root'@'5.15.0.3' IDENTIFIED BY PASSWORD
 '*47C562035B611FE7E0CD4CF0F58AEC9EDEF20E3' WITH GRANT OPTION;
```

Where:

- 5.15.0.3: this is the IP address of the remote Wildix system. Note that you need to execute the same command for all of the IPs you want to be able to use this database server. Specifically, all of the PBXs in the WMS network that are not the primary DB server (5.15.0.1 in our case).

2. '\*47C562035B611FE7E0CD4CF0F58AEC9EDEBF20E3': this is the hash of the root password. You can copy the value from the output of the command on step number 2, that way we will use the same password both for local and remote connections
- To apply changes immediately, execute following command on mysql cli:

```
>FLUSH PRIVILEGES;
```

- Now you can logout of mysql console by executing:

```
>quit
```

- There is one more step needed. By default mysql only accepts requests from the loopback. To finalize the enabling of remote connections, open the next config file with some text editor:

```
$vim /etc/mysql/mariadb.conf.d/50-server.cnf
```

- Search for a line with “bind-address = 127.0.0.1” and add a # to the beginning. This comments out this line and no longer restricts requests only to those that come on the loopback interface

```
#bind-address = 127.0.0.1
```

- Save and close the file, then execute:

```
$systemctl restart mysqld.service
```

Now we have our database server ready. It can be accessed by remote servers we added and on interfaces other than the loopback interface. You can now test it from the remote system (the other system part of the WMS Network) by executing:

```
$mysql -u root -p -h 5.15.0.1
```

And entering any password (invalid) you should see a message like the following:

```
root@jxxxxxxxxx:~# mysql -u root -p -h 5.15.0.1
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'5.15.0.3' (using password: YES)
root@jxxxxxxxxx:~# █
```

If you see the previous message, you can proceed to configuring the clients.

## Client configuration

Configuring the rest of the Wildix systems to use the server database is actually pretty easy (provided the server is ready). By default your Wildix system uses a local database. We just need to point to the remote server instead of the localhost. We accomplish this by editing the file `/etc/callweaver/cdr_mysql.conf`:

```

root@~:~# cat /etc/callweaver/cdr_mysql.conf
[global]
enabled = yes
hostname = 127.0.0.1
dbname = cdr
table = cdr
password = wil01dix
user = root
port = 3306
timeout = 2
charset = utf8mb4
root@~:~#
    
```

As you can see in the previous picture, Wildix is pointing to the localhost (127.0.0.1). You just need to change it to the server IP address, 5.15.0.1 in this example:

```

root@~:~# cat /etc/callweaver/cdr_mysql.conf
[global]
enabled = yes
;hostname = 127.0.0.1
hostname = 5.15.0.1
dbname = cdr
table = cdr
password = wil01dix
user = root
port = 3306
timeout = 10
charset = utf8mb4
root@~:~#
    
```

In the previous picture you can note that, besides the hostname, we also changed the timeout. This is a good idea since clients won't be using the localhost and surely the access to a remote DB will take more time than for a local DB.

Another thing worth mentioning is the fact that we did not change the dbname, table, user or port. This is because we are using a Wildix system as a database server, which means the database structure is ready. We did not change the password either because we configured the same password for the remote root access as the local root account was using.

### Result

Once the configuration is complete, you are able to see events for any system from any system:

	Date	From	Group	To	Group	Country	Call Class	Cost	Direction	Status	Talk time (s.)	Ring Time (s.)	PBX
1	01/07/2020 15:03	julio r (400)	Admin	eeee (401)	Default	Other	Internal	0.00	Local	Answ.	4	12	...
2	01/07/2020 15:02	julio r (400)	Admin	eeee (401)	Default	Other	Internal	0.00	Local	Answ.	6	11	...
3	01/07/2020 12:52	julio r (400)	Admin	eeee (401)	Default	Other	Internal	0.00	Local	Answ.	3	12	...

## Final Considerations

- If the systems you are working on already have some records in the CDR, all of the previous records on the clients won't be available on cdrview
- If you use a Wildix system as a database server, old records on the DB server cdrview will be available even after unifying CDRs
- If you are using an external database server, the database structure needs to be there before configuring the clients
- If you are using an external database server, no previous records will be available (unless a full dump was used to recreate the database. In such case, records contained in the dump will be available)
- If you are not using the same database name, table name, user or password, you need to adjust corresponding parameters on the cdr\_mysql.conf file