

# NDOUtils - Installing NDOUtils

Article Number: 406 | Rating: 3/5 from 2 votes | Last Updated: Mon, Jun 5, 2017 at 3:08 AM

## Overview

This document describes how to install NDOUtils from source.

This guide is broken up into several sections and covers different Linux distributions and operating systems. If your Linux Distribution or operating system is not included in this guide then develop the documentation.

NDOUtils 2.1.3 is what this guide instructs you to install, however future versions should also work fine with these steps.

**Note:** This guide is based on Nagios Core being installed using the following KB article:

[Documentation - Installing Nagios Core From Source](#)

You should always install the latest version of NDOUtils as Nagios Core will not start if an incompatible version is used. The most recent release of NDOUtils will always work with all v



**Nagios XI** includes NDOUtils and hence you **should NOT** follow this documentation.

Please select your OS:

- [Red Hat Enterprise Linux \(RHEL\)](#)
- [CentOS](#)
- [Oracle Linux](#)
- [Ubuntu](#)
- [SUSE SLES | openSUSE Leap](#)
- [Debian](#)
- [Raspbian](#)
- [Fedora](#)
- [Arch Linux](#)
- [Gentoo](#)
- [FreeBSD](#)
- [Solaris](#)
- [Apple OS X](#)

## RHEL | CentOS | Oracle Linux

### Prerequisites

Installing MySQL or MariaDB is required.

==== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
yum install -y mysql mysql-server mysql-devel
```

==== CentOS 7.x | RHEL 7.x =====

```
yum install -y mariadb mariadb-server mariadb-devel
```

==== Oracle Linux 7.x =====

```
cd /tmp
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
rpm -ivh mysql-community-release-el7-5.noarch.rpm
yum update
yum install -y mysql mysql-server mysql-devel perl-DBD-MySQL
```

### Start And Configure MySQL / MariaDB

Before configuring MySQL / MariaDB you must start the service and configure it to boot on startup.

==== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
service mysqld start
```

Check that it is running:

```
ps x | grep mysql | grep -v grep
```

Which should output something like:

```
1969 pts/0    S      0:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --socket=/var/lib/mysql/mysql.sock --pid-file=/var/run
```

Configure it to start when the system boots:

```
chkconfig --add mysqld
chkconfig mysqld on
```

==== CentOS 7.x | RHEL 7.x =====

```
systemctl start mariadb.service
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
2781 ?        Ss      0:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
2938 ?        Sl      0:00 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/v
```

Configure it to start when the system boots:

```
systemctl enable mariadb.service
```

==== Oracle Linux 7.x =====

```
systemctl start mysqld.service
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
1292 ?        Ss      0:00 /bin/sh /usr/bin/mysqld_safe
1760 ?        Sl      0:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/v
```

Configure it to start when the system boots:

```
systemctl enable mysqld.service
```

## Define MySQL / MariaDB Root Password

Now to define the password for the root account in MySQL / MariaDB.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

```
/usr/bin/mysqladmin -u root password 'mypassword'
```

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the 'mypassword'.

## Create Database

NDOUtils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (*we suggest you use a more secure password*).

The storage location of the database will be the default location that MySQL / MariaDB uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL / MariaDB database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press Enter after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL / MariaDB database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Linux Kernel Settings

NDOutils uses the kernel message queue for transferring the data from Nagios to NDOutils. We are going to increase the default values the Kernel boots with to ensure it operates o

First create a backup copy of the `/etc/sysctl.conf` file:

```
cp /etc/sysctl.conf /etc/sysctl.conf_backup
```

Now make the required changes:

```
sed -i '/msgmnb/d' /etc/sysctl.conf
sed -i '/msgmax/d' /etc/sysctl.conf
sed -i '/shmmax/d' /etc/sysctl.conf
sed -i '/shmall/d' /etc/sysctl.conf

printf "\n\nkernel.msgmnb = 131072000\n" >> /etc/sysctl.conf
printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.conf
printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.conf
printf "kernel.shmall = 268435456\n" >> /etc/sysctl.conf
sysctl -e -p /etc/sysctl.conf
```

The last command should output something like:

```
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOutils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/releases/download/ndoutils-2.1.3/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOutils

```
cd /tmp/ndoutils-2.1.3/
./configure
make all
```

## Install Binaries

This step installs the binary files.

```
make install
```

## Initialize Database

This prepares the database for NDOUtils.

```
cd db/
./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialized.

## Install Configuration Files

This installs the config files as well as configuring the MySQL / MariaDB credentials so NDOUtils can connect to the database.

The two config files are:

**/usr/local/nagios/etc/ndo2db.cfg**

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

**/usr/local/nagios/etc/ndomod.cfg**

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
make install-config
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

==== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
make install-init
```

==== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
make install-init
systemctl enable ndo2db.service
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

==== CentOS 5.x | RHEL 5.x | Oracle Linux 5.x =====

```
service ndo2db start
```

==== CentOS 6.x | RHEL 6.x | Oracle Linux 6.x =====

```
start ndo2db
```

==== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
systemctl start ndo2db.service
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cf
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

==== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
service nagios restart
service nagios status
```

==== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
systemctl restart nagios.service
systemctl status nagios.service
```

The last command should show Nagios running:

==== CentOS 5.x / 6.x | RHEL 5.x / 6.x | Oracle Linux 5.x / 6.x =====

```
nagios (pid 5345) is running...
```

==== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios)
   Active: active (running) since Tue 2016-10-04 12:31:00 AEDT; 22s ago
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11	
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11	
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status ndo2db.

===== CentOS 5.x | RHEL 5.x | Oracle Linux 5.x =====

```
service ndo2db start
service ndo2db stop
service ndo2db restart
service ndo2db status
```

===== CentOS 6.x | RHEL 6.x | Oracle Linux 6.x =====

```
start ndo2db
stop ndo2db
restart ndo2db
status ndo2db
```

===== CentOS 7.x | RHEL 7.x | Oracle Linux 7.x =====

```
systemctl start ndo2db.service
systemctl stop ndo2db.service
systemctl restart ndo2db.service
systemctl status ndo2db.service
```

## Ubuntu

### Prerequisites

Installing MySQL is required.

```
sudo apt-get update
sudo apt-get install -y mysql-server libmysqlclient-dev libdbd-mysql-perl
```

During the install you will be prompted for the password of the "root" user. This guide uses the password `mypassword` and will be used throughout this guide.

### Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the `'mypassword'`.

## Start And Configure MySQL

MySQL is started as part of the installation process and configured to start when the system boots:

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
8142 ?      Ssl      0:01 /usr/sbin/mysqld
```

## Create Database

NDOUTils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (we suggest you use a more secure password).

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (press **Enter** after each command):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
```

```
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
```

```
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Linux Kernel Settings

NDOUTils uses the kernel message queue for transferring the data from Nagios to NDOUTils. We are going to increase the default values the Kernel boots with to ensure it operates o

First create a backup copy of the `/etc/sysctl.conf` file:

```
sudo cp /etc/sysctl.conf /etc/sysctl.conf_backup
```

Now make the required changes:

```
sudo sed -i '/msgmnb/d' /etc/sysctl.conf
sudo sed -i '/msgmax/d' /etc/sysctl.conf
sudo sed -i '/shmmax/d' /etc/sysctl.conf
sudo sed -i '/shmall/d' /etc/sysctl.conf
sudo sh -c 'printf "\nkernel.msgmnb = 131072000\n" >> /etc/sysctl.conf'
sudo sh -c 'printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.conf'
sudo sh -c 'printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.conf'
sudo sh -c 'printf "kernel.shmall = 268435456\n" >> /etc/sysctl.conf'
sudo sysctl -e -p /etc/sysctl.conf
```

The last command should output something like:

```
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOUTils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOUTils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
```

```
cd /usr/local/nagios/etc/ndoutils
sudo ./configure
sudo make all
```

## Install Binaries

This step installs the binary files.

```
sudo make install
```

## Initialize Database

This prepares the database for NDOUtils.

```
cd db/
sudo ./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialized.

## Install Configuration Files

This installs the config files as well as configuring the MySQL credentials so NDOUtils can connect to the database.

The two config files are:

**/usr/local/nagios/etc/ndo2db.cfg**

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

**/usr/local/nagios/etc/ndomod.cfg**

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
sudo make install-config
sudo mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sudo sh -c 'sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg'
sudo sh -c 'sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg'
sudo mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

==== Ubuntu 13.x / 14.x ====

```
sudo make install-init
```

==== Ubuntu 15.x / 16.x / 17.x ====

```
sudo make install-init
sudo systemctl enable ndo2db.service
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

==== Ubuntu 13.x / 14.x ====

```
sudo service ndo2db start
```

==== Ubuntu 15.x / 16.x / 17.x ====

```
sudo systemctl start ndo2db.service
```



```
sudo systemctl start ndo2db.service
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
sudo sh -c 'printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg'
sudo sh -c 'printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/e
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

==== Ubuntu 13.x / 14.x =====

```
sudo service nagios restart
sudo service nagios status
```

==== Ubuntu 15.x / 16.x / 17.x =====

```
sudo systemctl restart nagios.service
sudo systemctl status nagios.service
```

The last command should show Nagios running:

==== Ubuntu 13.x / 14.x =====

```
nagios (pid 5345) is running...
```

==== Ubuntu 15.x / 16.x / 17.x =====

```
• nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios)
  Active: active (running) since Tue 2016-10-04 12:31:00 AEDT; 22s ago
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11	
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11	
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status ndo2db.

==== Ubuntu 13.x / 14.x ====

```
sudo start ndo2db
sudo stop ndo2db
sudo restart ndo2db
sudo status ndo2db
```

==== Ubuntu 15.x / 16.x / 17.x ====

```
sudo systemctl start ndo2db.service
sudo systemctl stop ndo2db.service
sudo systemctl restart ndo2db.service
sudo systemctl status ndo2db.service
```

## SUSE SLES | openSUSE Leap

### Prerequisites

Installing MySQL is required.

==== SUSE SLES 11.x ====

```
sudo zypper --non-interactive install mysql libmysqlclient-devel perl-DBD-mysql
```

==== SUSE SLES 12.x ====

```
cd /tmp
wget https://dev.mysql.com/get/mysql57-community-release-sles12-11.noarch.rpm
sudo rpm -ihv mysql57-community-release-sles12-11.noarch.rpm
sudo rpm --import /etc/RPM-GPG-KEY-mysql
sudo zypper refresh
```

Answer y to the signature

```
sudo zypper --non-interactive install mysql-community-server libmysqlclient-devel perl-DBD-mysql
```

==== openSUSE Leap 42.1 ====

```
sudo zypper --non-interactive install mysql-community-server libmysqlclient-devel perl-DBD-mysql
```

## Start And Configure MySQL

Before configuring MySQL you must start the service and configure it to boot as startup

before configuring MySQL you must start the service and configure it to boot on startup.

===== SUSE SLES 11.x =====

```
sudo /sbin/service mysql start
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
4319 pts/0    S      0:00 /bin/sh /usr/bin/mysqld_safe --mysqld=mysqld --user=mysql --pid-file=/var/run/mysql/mysqld.pid --socket=/var
4646 pts/0    S1     0:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --user=mysql -
```

Configure it to start when the system boots:

```
sudo /sbin/chkconfig --set mysql on
```

===== SUSE SLES 12.x | openSUSE Leap 42.1 =====

```
sudo systemctl start mysql.service
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
3461 ?        Ssl    0:00 /usr/sbin/mysqld --defaults-file=/etc/my.cnf --user=mysql
```

Configure it to start when the system boots:

```
sudo systemctl enable mysql.service
```

## Define MySQL Root Password

Now to define the password for the root account in MySQL.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

===== SUSE SLES 11.x | openSUSE Leap 42.1 =====

Execute the following command to define the password:

```
sudo /usr/bin/mysqladmin -u root password 'mypassword'
```

===== SUSE SLES 12.x =====

MySQL Community Edition creates a password for you by default and is recorded in the `/var/log/mysql/mysqld.log` file. To see what the password is type the following command

```
sudo grep 'temporary password' /var/log/mysql/mysqld.log
```

You should get an output like:

```
017-05-08T01:16:24.218861Z 1 [Note] A temporary password is generated for root@localhost: yURd-:G3Es(Z
```

To change the password, use the following command. Keep in mind the password needs to be a stronger password than the example of `mypassword` otherwise it will be rejected. Y

```
echo "SET Password FOR 'root'@'localhost' = PASSWORD('My@ssw0rd');" | mysql -u root -p'yURd-:G3Es(Z' --connect-expired-password
```

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the 'mypassword'.

## Create Database

NDOUTils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (*we suggest you use a more secure password ... MySQL Community Edition w*

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press **Enter** after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION;
```

Now you can exit the local MySQL database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Linux Kernel Settings

NDOutils uses the kernel message queue for transferring the data from Nagios to NDOutils. We are going to increase the default values the Kernel boots with to ensure it operates o

First create a backup copy of the `/etc/sysctl.conf` file:

```
sudo cp /etc/sysctl.conf /etc/sysctl.conf_backup
```

Now make the required changes:

```
sudo sed -i '/msgmnb/d' /etc/sysctl.conf
sudo sed -i '/msgmax/d' /etc/sysctl.conf
sudo sed -i '/shmmax/d' /etc/sysctl.conf
sudo sed -i '/shmall/d' /etc/sysctl.conf
sudo sh -c 'printf "\nkernel.msgmnb = 131072000\n" >> /etc/sysctl.conf'
sudo sh -c 'printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.conf'
sudo sh -c 'printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.conf'
sudo sh -c 'printf "kernel.shmall = 268435456\n" >> /etc/sysctl.conf'
```

Then this last command depends on the OS version you are running:

==== SUSE SLES 11x ====

```
sudo /sbin/sysctl -e -p /etc/sysctl.conf
```

==== SUSE SLES 12.x | openSUSE Leap ====

```
sudo /usr/sbin/sysctl -e -p /etc/sysctl.conf
```

The last command should output something like:

```
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOutils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOutils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
```

```
sudo ./configure
sudo make all
```

## Install Binaries

This step installs the binary files.

```
sudo make install
```

## Initialize Database

This prepares the database for NDOUtils.

==== SUSE SLES 11x ====

```
cd db/
sudo ./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

==== SUSE SLES 12.x | openSUSE Leap ====

```
cd db/
sudo ./installdb -u 'ndoutils' -p 'ndoutils_password' -h '127.0.0.1' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialized.

## Install Configuration Files

This installs the config files as well as configuring the MySQL credentials so NDOUtils can connect to the database.

The two config files are:

**/usr/local/nagios/etc/ndo2db.cfg**

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

For SUSE SLES 12.x you will also need to change `db_host=localhost` to:

```
db_host=127.0.0.1
```

**/usr/local/nagios/etc/ndomod.cfg**

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
sudo make install-config
sudo mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sudo sh -c 'sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg'
sudo sh -c 'sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg'
sudo mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

For SUSE SLES 12.x you will also need to change `db_host=localhost` to:

```
sudo sh -c 'sed -i 's/^db_host=.*db_host=127\.\0\.\0\.\1/g' /usr/local/nagios/etc/ndo2db.cfg'
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

==== SUSE SLES 11x ====

```
sudo make install-init
sudo /sbin/chkconfig --set ndo2db on
```

==== SUSE SLES 12.x | openSUSE ====

```
sudo make install-init
sudo systemctl enable ndo2db.service
```

```
sudo systemctl enable ndo2db.service
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

===== SUSE SLES 11.x =====

```
sudo /sbin/service ndo2db start
```

===== SUSE SLES 12.x | openSUSE =====

```
sudo systemctl start ndo2db.service
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
sudo sh -c 'printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg'
sudo sh -c 'printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/e
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

===== SUSE SLES 11.x =====

```
sudo /sbin/service nagios restart
sudo /sbin/service nagios status
```

===== SUSE SLES 12.x | openSUSE =====

```
sudo systemctl restart nagios.service
sudo systemctl status nagios.service
```

The last command should show Nagios running:

===== SUSE SLES 11.x =====

```
nagios (pid 5345) is running...
```

===== SUSE SLES 12.x | openSUSE =====

```
• nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios)
  Active: active (running) since Tue 2016-10-04 12:31:00 AEDT; 22s ago
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
```

```
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11	
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11	
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status NRPE.

===== SUSE SLES 11x =====

```
sudo /sbin/service ndo2db start
sudo /sbin/service ndo2db stop
sudo /sbin/service ndo2db restart
sudo /sbin/service ndo2db status
```

===== SUSE SLES 12.x | openSUSE Leap 42.1 =====

```
sudo systemctl start ndo2db.service
sudo systemctl stop ndo2db.service
sudo systemctl restart ndo2db.service
sudo systemctl status ndo2db.service
```

## Debian | Raspbian

All steps on Debian require to run as root. To become root simply run:

Debian:

```
su
```

Raspbian:

```
sudo -i
```

All commands from this point onwards will be as root.

All commands from this point onwards will be as root.

## Prerequisites

Installing MySQL is required.

```
apt-get update
apt-get install -y mysql-server libmysqlclient-dev libdbd-mysql-perl
```

During the install you will be prompted for the password of the "root" user. This guide uses the password **mypassword** and will be used throughout this guide.

## Start And Configure MySQL

MySQL is started as part of the installation process and configured to start when the system boots:

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
8338 ?        S          0:00 /bin/sh /usr/bin/mysqld_safe
8686 ?        Sl         0:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/plugin --user=mysql --l
```

## Define MySQL Root Password

Now to define the password for the root account in MySQL.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters. This was already do

### Password Note:

In future commands you will see the password is provided using the **-p** argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the **-p** and the 'mypassword'.

## Create Database

NDOUTils requires a database to be created which will be called **nagios**.

There will also be a dedicated user account called **ndoutils** with the password **ndoutils\_password** (*we suggest you use a more secure password*).

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press Enter after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
```

```
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
```

```
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Linux Kernel Settings

NDOUTils uses the kernel message queue for transferring the data from Nagios to NDOUTils. We are going to increase the default values the Kernel boots with to ensure it operates o

First create a backup copy of the `/etc/sysctl.conf` file:



first create a backup copy of the `/etc/sysctl.conf` file:

```
cp /etc/sysctl.conf /etc/sysctl.conf_backup
```

Now make the required changes:

```
sed -i '/msgmnb/d' /etc/sysctl.conf
sed -i '/msgmax/d' /etc/sysctl.conf
sed -i '/shmmax/d' /etc/sysctl.conf
sed -i '/shmall/d' /etc/sysctl.conf
printf "\nkernel.msgmnb = 131072000\n" >> /etc/sysctl.conf
printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.conf
printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.conf
printf "kernel.shmall = 268435456\n" >> /etc/sysctl.conf
sysctl -e -p /etc/sysctl.conf
```

The last command should output something like:

```
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOutils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOutils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
./configure
make all
```

## Install Binaries

This step installs the binary files.

```
make install
```

## Initialize Database

This prepares the database for NDOutils.

```
cd db/
./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialized.

## Install Configuration Files

This installs the config files as well as configuring the MySQL credentials so NDOutils can connect to the database.

The two config files are:

```
/usr/local/nagios/etc/ndo2db.cfg
```

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

You should make sure the username and password are correct.

```
/usr/local/nagios/etc/ndomod.cfg
```

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
make install-config
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

==== Debian 7.x ====

```
make install-init
update-rc.d ndo2db defaults
```

==== Debian 8.x ====

```
make install-init
systemctl enable ndo2db.service
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

==== Debian 7.x ====

```
service ndo2db start
```

==== Debian 8.x ====

```
systemctl start ndo2db.service
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cf
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

==== Debian 7.x ====

```
service nagios restart
service nagios status
```

==== Debian 8.x ====

```
systemctl restart nagios.service
systemctl status nagios.service
```

The last command should show Nagios running:

==== Debian 7.x ====

```
nagios (pid 5345) is running...
```

==== Debian 8.x ====

```
● nagios.service - LSB: Starts and stops the Nagios monitoring server
   Loaded: loaded (/etc/rc.d/init.d/nagios)
   Active: active (running) since Tue 2016-10-04 12:31:00 AEDT; 22s ago
```

## Check NDOutils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1		
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1		
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1		
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1		
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1		
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1		
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11		
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11		
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11		
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1		
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1		
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1		
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1		
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1		
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1		
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1		
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1		
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1		
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11		
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1		
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1		
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1		
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1		
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1		
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' ini			
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026			

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status ndo2db.

==== Debian 7.x =====

```
service ndo2db start
service ndo2db stop
service ndo2db restart
service ndo2db status
```

===== Debian 8.x =====

```
systemctl start ndo2db.service
systemctl stop ndo2db.service
systemctl restart ndo2db.service
systemctl status ndo2db.service
```

## Fedora

### Prerequisites

Installing MariaDB is required.

```
dnf install -y mariadb mariadb-server mariadb-devel
dnf update -y
```

### Start And Configure MariaDB

Before configuring MariaDB you must start the service and configure it to boot on startup.

```
systemctl start mariadb.service
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
2781 ?      Ss      0:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
2938 ?      Sl      0:00 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=
```

Configure it to start when the system boots:

```
systemctl enable mariadb.service
```

### Define MySQL / MariaDB Root Password

Now to define the password for the root account in MySQL / MariaDB.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

```
/usr/bin/mysqladmin -u root password 'mypassword'
```

### Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the 'mypassword'.

### Create Database

NDOUtils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (*we suggest you use a more secure password*).

The storage location of the database will be the default location that MySQL / MariaDB uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL / MariaDB database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press **Enter** after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL / MariaDB database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Linux Kernel Settings

NDOUTils uses the kernel message queue for transferring the data from Nagios to NDOUTils. We are going to increase the default values the Kernel boots with to ensure it operates o

First create a backup copy of the `/etc/sysctl.conf` file:

```
cp /etc/sysctl.conf /etc/sysctl.conf_backup
```

Now make the required changes:

```
sed -i '/msgmnb/d' /etc/sysctl.conf
sed -i '/msgmax/d' /etc/sysctl.conf
sed -i '/shmmax/d' /etc/sysctl.conf
sed -i '/shmall/d' /etc/sysctl.conf
printf "\n\kernel.msgmnb = 131072000\n" >> /etc/sysctl.conf
printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.conf
printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.conf
printf "kernel.shmall = 268435456\n" >> /etc/sysctl.conf
sysctl -e -p /etc/sysctl.conf
```

The last command should output something like:

```
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOUTils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOUTils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
./configure
make all
```

## Install Binaries

This step installs the binary files.

```
make install
```

## Initialize Database

This prepares the database for NDOUTils.

```
cd db/
./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command "make install" will do the following:

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installldb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialized.

## Install Configuration Files

This installs the config files as well as configuring the MySQL / MariaDB credentials so NDOUtils can connect to the database.

The two config files are:

```
/usr/local/nagios/etc/ndo2db.cfg
```

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

```
/usr/local/nagios/etc/ndomod.cfg
```

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
make install-config
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

```
make install-init
systemctl enable ndo2db.service
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

```
systemctl start ndo2db.service
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cfg
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nagios service.

```
systemctl restart nagios.service
systemctl status nagios.service
```

The last command should show Nagios running:

```
• nagios.service - LSB: Starts and stops the Nagios monitoring server
  Loaded: loaded (/etc/rc.d/init.d/nagios)
  Active: active (running) since Tue 2016-10-04 12:31:00 AEDT; 22s ago
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11	
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11	
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status ndo2db.

```
systemctl start ndo2db.service
systemctl stop ndo2db.service
systemctl restart ndo2db.service
systemctl status ndo2db.service
```

## Prerequisites

Installing MariaDB is required.

```
pacman --noconfirm -Syuu  
pacman --noconfirm -S mariadb perl-dbi perl-dbd-mysql  
mysql_install_db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

## Start And Configure MariaDB

```
systemctl start mariadb.service
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
60211 ?        Ssl      0:00 /usr/bin/mysqld
```

Configure it to start when the system boots:

```
systemctl enable mariadb.service
```

## Define MariaDB Root Password

Now to define the password for the root account in MariaDB.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

```
/usr/bin/mysqladmin -u root password 'mypassword'
```

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the 'mypassword'.

## Create Database

NDOUTILS requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (*we suggest you use a more secure password*).

The storage location of the database will be the default location that MariaDB uses, this can be changed however it is not covered in this guide.

This command will connect to the local MariaDB database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press **Enter** after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;  
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';  
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M  
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MariaDB database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database  
information_schema  
nagios
```



```
test
```

## Linux Kernel Settings

NDOUtils uses the kernel message queue for transferring the data from Nagios to NDOUtils. We are going to increase the default values the Kernel boots with to ensure it operates o  
In a fresh install of Arch Linux the `/etc/sysctl.d/99-sysctl.conf` file exists:

These commands make the required changes:

```
printf "\n\kernel.msgmnb = 131072000\n" >> /etc/sysctl.d/99-sysctl.conf
printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.d/99-sysctl.conf
printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.d/99-sysctl.conf
printf "kernel.shmall = 268435456\n" >> /etc/sysctl.d/99-sysctl.conf
sysctl -e -p /etc/sysctl.d/99-sysctl.conf
```

The last command should output something like:

```
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOUtils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOUtils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
./configure
make all
```

## Install Binaries

This step installs the binary files.

```
make install
```

## Initialize Database

This prepares the database for NDOUtils.

```
cd db/
./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initiali

## Install Configuration Files

This installs the config files as well as configuring the MariaDB credentials so NDOUtils can connect to the database.

The two config files are:

```
/usr/local/nagios/etc/ndo2db.cfg
```

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

```
/usr/local/nagios/etc/ndomod.cfg
```

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
make install-config
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

```
make install-init
systemctl enable ndo2db.service
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

```
systemctl start ndo2db.service
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cf
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

```
systemctl restart nagios.service
systemctl status nagios.service
```

The last command should show Nagios running:

```
• nagios.service - Nagios Monitoring System
  Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor preset: disabled)
  Active: active (running) since Fri 2017-05-05 19:56:41 UTC; 6s ago
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1494014201] ndomod: NDOMOD 2.1.3 (2017-04-13) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1494014201] ndomod: Could not open data sink! I'll keep trying, but some output may get lost...
[1494014201] ndomod registered for process data
[1494014201] ndomod registered for timed event data
[1494014201] ndomod registered for log data'
[1494014201] ndomod registered for system command data'
[1494014201] ndomod registered for event handler data'
[1494014201] ndomod registered for notification data'
[1494014201] ndomod registered for service check data'
[1494014201] ndomod registered for host check data'
[1494014201] ndomod registered for comment data'
[1494014201] ndomod registered for downtime data'
[1494014201] ndomod registered for flapping data'
[1494014201] ndomod registered for program status data'
[1494014201] ndomod registered for host status data'
[1494014201] ndomod registered for service status data'
[1494014201] ndomod registered for adaptive program data'
[1494014201] ndomod registered for adaptive host data'
[1494014201] ndomod registered for adaptive service data'
[1494014201] ndomod registered for external command data'
[1494014201] ndomod registered for aggregated status data'
[1494014201] ndomod registered for retention data'
[1494014201] ndomod registered for contact data'
```

```
[1494014201] ndomod registered for contact notification data'
[1494014201] ndomod registered for acknowledgement data'
[1494014201] ndomod registered for state change data'
[1494014201] ndomod registered for contact status data'
[1494014201] ndomod registered for adaptive contact data'
[1494014201] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326176	262144	ndomod registered for log data'	1 1	
2	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326186	262144	ndomod registered for system command data'	1 1	
3	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326191	262144	ndomod registered for event handler data'	1 1	
4	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326196	262144	ndomod registered for notification data'	1 1	
5	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326201	262144	ndomod registered for service check data'	1 1	
6	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326205	262144	ndomod registered for host check data'	1 1	
7	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326210	262144	ndomod registered for comment data'	1 1	
8	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326215	262144	ndomod registered for downtime data'	1 1	
9	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326220	262144	ndomod registered for flapping data'	1 1	
10	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326225	262144	ndomod registered for program status data'	1 1	
11	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326230	262144	ndomod registered for host status data'	1 1	
12	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326234	262144	ndomod registered for service status data'	1 1	
13	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326239	262144	ndomod registered for adaptive program data'	1 1	
14	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326243	262144	ndomod registered for adaptive host data'	1 1	
15	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326254	262144	ndomod registered for adaptive service data'	1 1	
16	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326260	262144	ndomod registered for external command data'	1 1	
17	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326264	262144	ndomod registered for aggregated status data'	1 1	
18	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326268	262144	ndomod registered for retention data'	1 1	
19	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326273	262144	ndomod registered for contact data'	1 1	
20	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326277	262144	ndomod registered for contact notification data'	1 1	
21	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326282	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326287	262144	ndomod registered for state change data'	1 1	
23	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326291	262144	ndomod registered for contact status data'	1 1	
24	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326296	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326301	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2017-05-05 19:56:41	2017-05-05 19:56:41	812884	262144	Successfully launched command file worker with pid 3995		

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status ndo2db.

```
systemctl start ndo2db.service
systemctl stop ndo2db.service
systemctl restart ndo2db.service
systemctl status ndo2db.service
```

## Gentoo

### Prerequisites

Installing MariaDB is required.

```
emerge --sync
emerge --noreplace dev-db/mysql
```

### Define MySQL Root Password

The output of the installation of mysql will produce some thing like:

```
* MySQL MY_DATADIR is /var/lib/mysql
* You might want to run:
* "emerge --config =dev-db/mysql-5.6.35"
* if this is a new install.
```

You will need to execute the command:

```
emerge --config =dev-db/mysql-5.6.35
```

This will prompt you to type a password for the root account in MySQL.

This KB article is using the password of 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

Once you've provided the password MySQL will be configured.

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the `'mypassword'`.

## Start And Configure MySQL

```
service mysql start
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
28862 ?        Ssl      0:00 /usr/sbin/mysqld --defaults-file=/etc/mysql/my.cnf
```

Configure it to start when the system boots:

```
rc-update add mysql default
```

## Create Database

NDOUTils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (we suggest you use a more secure password).

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (press **Enter** after each command):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
```

```
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
```

```
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MariaDB database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Linux Kernel Settings

NDOUTils uses the kernel message queue for transferring the data from Nagios to NDOUTils. We are going to increase the default values the Kernel boots with to ensure it operates o

In a fresh install of Arch Linux the `/etc/sysctl.conf` file exists:

These commands make the required changes:

```
printf "\n\nkernel.msgmnb = 131072000\n" >> /etc/sysctl.conf
printf "kernel.msgmax = 131072000\n" >> /etc/sysctl.conf
printf "kernel.shmmax = 4294967295\n" >> /etc/sysctl.conf
printf "kernel.shmall = 268435456\n" >> /etc/sysctl.conf
sysctl -e -p /etc/sysctl.conf
```

The last command should output something like:

```
kernel.msgmnb = 131072000
kernel.msgmax = 131072000
```

```
kernel.msgmax = 131072000
kernel.shmmax = 4294967295
kernel.shmall = 268435456
```

There is no need to reboot the system, the last command ensured the new settings are active in the kernel.

## Downloading NDOutils Source

Some additional files will be downloaded so the init scripts are correctly created.

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
wget --no-check-certificate -O ndoutils-ndoutils-2.1.3/startup/gentoo-init.in https://raw.githubusercontent.com/box293/ndoutils/master/
wget --no-check-certificate -O ndoutils-ndoutils-2.1.3/Makefile.in https://raw.githubusercontent.com/box293/ndoutils/master/Makefile.in
wget --no-check-certificate -O ndoutils-ndoutils-2.1.3/configure https://raw.githubusercontent.com/box293/ndoutils/master/configure
```

## Compile NDOutils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
./configure
make all
```

## Install Binaries

This step installs the binary files.

```
make install
```

## Initialize Database

This prepares the database for NDOutils.

```
cd db/
./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initiali

## Install Configuration Files

This installs the config files as well as configuring the MariaDB credentials so NDOutils can connect to the database.

The two config files are:

```
/usr/local/nagios/etc/ndo2db.cfg
```

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

```
/usr/local/nagios/etc/ndomod.cfg
```

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
make install-config
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
sed -i 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

```
make install-init
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

```
service ndo2db start
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cf
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

```
service nagios restart
service nagios status
```

The last command should show Nagios running:

```
* Executing: /lib/rc/sh/openrc-run.sh /lib/rc/sh/openrc-run.sh /etc/init.d/nagios status
* status: started
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1494014201] ndomod: NDOMOD 2.1.3 (2017-04-13) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1494014201] ndomod: Could not open data sink! I'll keep trying, but some output may get lost...
[1494014201] ndomod registered for process data
[1494014201] ndomod registered for timed event data
[1494014201] ndomod registered for log data'
[1494014201] ndomod registered for system command data'
[1494014201] ndomod registered for event handler data'
[1494014201] ndomod registered for notification data'
[1494014201] ndomod registered for service check data'
[1494014201] ndomod registered for host check data'
[1494014201] ndomod registered for comment data'
[1494014201] ndomod registered for downtime data'
[1494014201] ndomod registered for flapping data'
[1494014201] ndomod registered for program status data'
[1494014201] ndomod registered for host status data'
[1494014201] ndomod registered for service status data'
[1494014201] ndomod registered for adaptive program data'
[1494014201] ndomod registered for adaptive host data'
[1494014201] ndomod registered for adaptive service data'
[1494014201] ndomod registered for external command data'
[1494014201] ndomod registered for aggregated status data'
[1494014201] ndomod registered for retention data'
[1494014201] ndomod registered for contact data'
[1494014201] ndomod registered for contact notification data'
[1494014201] ndomod registered for acknowledgement data'
[1494014201] ndomod registered for state change data'
[1494014201] ndomod registered for contact status data'
[1494014201] ndomod registered for adaptive contact data'
[1494014201] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326176	262144	ndomod registered for log data'	11	
2	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326186	262144	ndomod registered for system command data'	1	1

2	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326186	262144	ndomod registered for system command data'	1	1
3	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326191	262144	ndomod registered for event handler data'	1	1
4	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326196	262144	ndomod registered for notification data'	1	1
5	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326201	262144	ndomod registered for service check data'	1	1
6	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326205	262144	ndomod registered for host check data'	1	1
7	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326210	262144	ndomod registered for comment data'	1	1
8	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326215	262144	ndomod registered for downtime data'	1	1
9	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326220	262144	ndomod registered for flapping data'	1	1
10	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326225	262144	ndomod registered for program status data'	1	1
11	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326230	262144	ndomod registered for host status data'	1	1
12	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326234	262144	ndomod registered for service status data'	1	1
13	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326239	262144	ndomod registered for adaptive program data'	1	1
14	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326243	262144	ndomod registered for adaptive host data'	1	1
15	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326254	262144	ndomod registered for adaptive service data'	1	1
16	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326260	262144	ndomod registered for external command data'	1	1
17	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326264	262144	ndomod registered for aggregated status data'	1	1
18	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326268	262144	ndomod registered for retention data'	1	1
19	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326273	262144	ndomod registered for contact data'	1	1
20	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326277	262144	ndomod registered for contact notification data'	1	1
21	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326282	262144	ndomod registered for acknowledgement data'	1	1
22	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326287	262144	ndomod registered for state change data'	1	1
23	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326291	262144	ndomod registered for contact status data'	1	1
24	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326296	262144	ndomod registered for adaptive contact data'	1	1
25	1	2017-05-05 19:56:41	2017-05-05 19:56:41	326301	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2017-05-05 19:56:41	2017-05-05 19:56:41	812884	262144	Successfully launched command file worker with pid 3995		

## Service Commands

Different Linux distributions have different methods of starting / stopping / restarting / status ndo2db.

```
service ndo2db start
service ndo2db stop
service ndo2db restart
service ndo2db status
```

## FreeBSD



There is an issue with NDOUtils on FreeBSD that have been reported in the following GitHub issue:

[Module ndomod.o is using an old or unspecified version of the event broker API. Module will be unloaded.](#)

## Prerequisites

Installing MySQL is required.

In the steps below, when installing FreeBSD packages you will be prompted with screens asking what you would like installed. You can just press **Enter** to accept the default selection.

```
pkg install -y mysql56-server p5-DBD-mysql p5-DBI
```

## Start And Configure MySQL

Before configuring MySQL you must start the service and configure it to boot on startup.

```
echo 'mysql_enable="YES"' >> /etc/rc.conf
service mysql-server start
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
80262 - Ss    0:00.05 /bin/sh /usr/local/bin/mysqld_safe --defaults-extra-file=/var/db/mysql/my.cnf --user=mysql -
80362 - S    0:00.28 /usr/local/libexec/mysqld --defaults-extra-file=/var/db/mysql/my.cnf --basedir=/usr/local --
```

MySQL has been configured to start when the system boots.

## Define MySQL Root Password

Now to define the password for the root account in MySQL.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

To change the password, use the following command:

```
mysqladmin -u root password 'mypassword'
```

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the `'mypassword'`.

## Create Database

NDOutils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (we suggest you use a more secure password).

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (press **Enter** after each command):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
```

```
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
```

```
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Operating System Kernel Settings

NDOutils uses the kernel message queue for transferring the data from Nagios to NDOutils. We are going to increase the default values the Kernel boots with to ensure it operates o

First create a backup copy of the `/etc/sysctl.conf` and `/boot/loader.conf` files:

```
cp /etc/sysctl.conf /etc/sysctl.conf_backup
cp /boot/loader.conf /boot/loader.conf_backup
```

Now make the required changes:

```
sed -i '' '/msgmnb/d' /boot/loader.conf
sed -i '' '/msgmax/d' /boot/loader.conf
sed -i '' '/shmmax/d' /etc/sysctl.conf
sed -i '' '/shmall/d' /etc/sysctl.conf
printf "\n\kern.ipc.msgmnb = 131072000\n" >> /boot/loader.conf
printf "kern.ipc.msgmax = 131072000\n" >> /boot/loader.conf
printf "\n\kern.ipc.shmmax = 4294967295\n" >> /etc/sysctl.conf
printf "kern.ipc.shmall = 268435456\n" >> /etc/sysctl.conf
```

You need to reboot the system to ensure the new settings are active in the kernel:

```
reboot
```

## Downloading NDOutils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```



## Compile NDOUtils

```
cd /tmp/ndoutils-ndoutils-2.1.3/  
./configure  
gmake all
```

## Install Binaries

This step installs the binary files.

```
gmake install
```

## Initialize Database

This prepares the database for NDOUtils.

```
cd db/  
perl ./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios  
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.  
** Creating tables for version 2.0.1  
    Using mysql.sql for installation...  
** Updating table nagios_dbversion  
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialized.

## Install Configuration Files

This installs the config files as well as configuring the MySQL credentials so NDOUtils can connect to the database.

The two config files are:

**/usr/local/nagios/etc/ndo2db.cfg**

The following lines are defined in this file:

```
db_user=ndoutils  
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

**/usr/local/nagios/etc/ndomod.cfg**

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
gmake install-config  
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg  
sed -i 's/^db_user=.*\/db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg  
sed -i 's/^db_pass=.*\/db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg  
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

```
gmake install-init  
echo '/usr/local/etc/rc.d/ndo2db start' >> /etc/rc.local
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

This command starts the ndo2db service.

```
service ndo2db start
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the **nagios.cfg** file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
```

```
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cf
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

```
service nagios restart
service nagios status
```

The last command should show Nagios running:

```
nagios (pid 5345) is running...
```

## Check NDOutils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11	
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11	
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	

24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1	1
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

These commands are for starting / stopping / restarting / status ndo2db.

```
service ndo2db start
service ndo2db stop
service ndo2db restart
service ndo2db status
```

## Solaris

There are some issues with NDOutils on Solaris that have been reported in the following GitHub issues:



[ndo.sock is not deleted when service is disabled/stopped](#)

[cannot get ndo2db service to start when the -f argument is present](#)

[ndo2db service will not start if nagios user home directory does not exist](#)

## Prerequisites

Installing MySQL is required.

```
pkg install --accept mysql-56 mysql-56/library mysql-56/client perl-5/dbd-mysql
pkgutil -i libmysqlclient18 mysql_dev pm_dbd_mysql pm_dbi
echo 'export PATH=$PATH:/usr/mysql/5.6/bin' >> ~/.profile
source ~/.profile
mkdir /export/home/nagios
```

## Start And Configure MySQL

Before configuring MySQL you must start the service and configure it to boot on startup.

```
svcadm enable mysql
```

Check that it is running:

```
ps ax | grep mysql | grep -v grep
```

Which should output something like:

```
3276 ?      S  0:00 /bin/sh /usr/mysql/5.6/bin/mysqld_safe --defaults-file=
3352 ?      S  0:00 /usr/mysql/5.6/bin/mysqld --defaults-file=/etc/mysql/5.
```

MySQL has been configured to start when the system boots.

## Define MySQL Root Password

Now to define the password for the root account in MySQL.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

```
/usr/bin/mysqladmin -u root password 'mypassword'
```

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the `'mypassword'`.

## Create Database

NDOutils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (*we suggest you use a more secure password*).

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press **Enter** after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Kernel Settings

NDOUtils uses the kernel message queue for transferring the data from Nagios to NDOUtils.

Message queues have changed in Solaris as per this documentation:

<http://docs.oracle.com/cd/E19253-01/817-0404/appendix-6/index.html>

<http://docs.oracle.com/cd/E19253-01/817-1592/rmctrls-1/index.html>

This documentation does not provide any steps on adjusting the interprocess message message queue parameters (as seen in the other operating systems in this documentation). I

## Downloading NDOUtils Source

```
cd /tmp
wget -O ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOUtils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
./configure
gmake all
```

## Install Binaries

This step installs the binary files.

```
gmake install
```

## Initialize Database

This prepares the database for NDOUtils.

```
cd db/
/opt/csw/bin/perl ./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initialia

## Install Configuration Files

This installs the config files as well as configuring the MySQL credentials so NDOUtils can connect to the database.

The two config files are:

`/usr/local/nagios/etc/ndo2db.cfg`

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

`/usr/local/nagios/etc/ndomod.cfg`

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
gmake install-config
mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
perl -p -i -e 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
perl -p -i -e 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```

## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

```
perl -p -i -e 's/cfg -f/cfg/g' startup/solaris-init.xml
gmake install-init
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

```
svcadm enable ndo2db
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg
printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/etc/nagios.cf
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

```
svcadm disable nagios
svcadm enable nagios
svcs nagios
```

The last command should show Nagios running:

```
STATE      STIME      FMRI
online     15:42:02  svc:/application/nagios:default
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data
```

```
[1475544660] naomoda registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	instance_time	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'	11	
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'	11	
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'	1 1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o' initi		
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

These commands are for starting / stopping / restarting / status ndo2db.

```
svcadm enable ndo2db
svcadm disable ndo2db
svcadm restart ndo2db
svcs ndo2db
svcs -xv ndo2db
```

The following KB article has important information on Solaris and services:

[How To Clear Solaris Service Maintenance Status](#)

## Apple OS X



There is an issue with NDOUtils on Apple OS X that have been reported in the following GitHub issue:

[Symbol not found: \\_\\_nagios\\_object\\_structure\\_version](#)

## Prerequisites

Installing MySQL is required.

This assumes that Xcode and MacPorts is already installed, which is detailed in this Nagios Core install guide:

### [Installing Nagios Core From Source](#)

Once you've done this, follow these steps in a terminal session:

```
sudo port install mysql56-server
sudo port select mysql mysql56
sudo port install p5-dbd-mysql
```

## Start And Configure MySQL

Before configuring MySQL you must start the service and configure it to boot on startup.

```
sudo -u _mysql mysql_install_db
sudo chown -R _mysql:_mysql /opt/local/var/db/mysql56/
sudo chown -R _mysql:_mysql /opt/local/var/run/mysql56/
sudo chown -R _mysql:_mysql /opt/local/var/log/mysql56/
sudo port load mysql56-server
```

Check that it is running:

```
ps -ax | grep mysql | grep -v grep
```

Which should output something like:

```
573 ??          0:00.51 /opt/local/lib/mysql56/bin/mysqld
```

MySQL has been configured to start when the system boots.

## Define MySQL Root Password

Now to define the password for the root account in MySQL.

The password being defined is 'mypassword' and will be used in future commands (*we suggest you use a more secure password*).

The 'single quotes' are used to define the boundaries of the password, this is extremely important when the password contains a space or special characters.

```
/opt/local/bin/mysqladmin -u root password 'mypassword'
```

## Password Note:

In future commands you will see the password is provided using the `-p` argument like follows:

```
mysql -u root -p'mypassword'
```

**NOTE:** It's very important to **NOT** put a space between the `-p` and the 'mypassword'.

## Create Database

NDOutils requires a database to be created which will be called `nagios`.

There will also be a dedicated user account called `ndoutils` with the password `ndoutils_password` (*we suggest you use a more secure password*).

The storage location of the database will be the default location that MySQL uses, this can be changed however it is not covered in this guide.

This command will connect to the local MySQL database engine interface.

```
mysql -u root -p'mypassword'
```

Now execute these four commands (*press **Enter** after each command*):

```
CREATE DATABASE nagios DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE USER 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password';
```

```
GRANT USAGE ON *.* TO 'ndoutils'@'localhost' IDENTIFIED BY 'ndoutils_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
```

```
GRANT ALL PRIVILEGES ON nagios.* TO 'ndoutils'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u ndoutils -p'ndoutils_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagios
test
```

## Kernel Settings

NDOUtils uses the kernel message queue for transferring the data from Nagios to NDOUtils. This documentation does not provide any steps on adjusting the interprocess message r feel free to contact us.

## Downloading NDOUtils Source

```
cd /tmp
curl -L -o ndoutils.tar.gz https://github.com/NagiosEnterprises/ndoutils/archive/ndoutils-2.1.3.tar.gz
tar xzf ndoutils.tar.gz
```

## Compile NDOUtils

```
cd /tmp/ndoutils-ndoutils-2.1.3/
sudo ./configure
sudo make all
```

## Install Binaries

This step installs the binary files.

```
sudo make install
```

## Initialize Database

This prepares the database for NDOUtils.

```
cd db/
sudo perl ./installdb -u 'ndoutils' -p 'ndoutils_password' -h 'localhost' -d nagios
cd ..
```

The command will produce output similar to the following:

```
DBD::mysql::db do failed: Table 'nagios.nagios_dbversion' doesn't exist at ./installdb line 52.
** Creating tables for version 2.0.1
    Using mysql.sql for installation...
** Updating table nagios_dbversion
Done!
```

That first line of output that says **failed: Table 'nagios.nagios\_dbversion' doesn't exist** is expected, it's testing to make sure the database hasn't already been initiali

## Install Configuration Files

This installs the config files as well as configuring the MySQL credentials so NDOUtils can connect to the database.

The two config files are:

**/usr/local/nagios/etc/ndo2db.cfg**

The following lines are defined in this file:

```
db_user=ndoutils
db_pass=ndoutils_password
```

You should make sure the username and password are correct.

**/usr/local/nagios/etc/ndomod.cfg**

No changes are required in this file.

Using the default username and password the following commands install the configuration files and make the required changes.

```
sudo make install-config
sudo mv /usr/local/nagios/etc/ndo2db.cfg-sample /usr/local/nagios/etc/ndo2db.cfg
sudo sed -i '' 's/^db_user=.*db_user=ndoutils/g' /usr/local/nagios/etc/ndo2db.cfg
sudo sed -i '' 's/^db_pass=.*db_pass=ndoutils_password/g' /usr/local/nagios/etc/ndo2db.cfg
sudo mv /usr/local/nagios/etc/ndomod.cfg-sample /usr/local/nagios/etc/ndomod.cfg
```



## Install Service / Daemon

This installs the service or daemon files and configure them to start on boot.

```
sudo make install-init
```

Information on starting and stopping services will be explained further on.

## Start Service / Daemon

Different Linux distributions have different methods of starting the ndo2db service.

```
sudo systemctl start org.nagios.ndo2db
```

## Update Nagios To Use NDO Broker Module

Now you need to tell Nagios to use the NDO broker module. This is as simple as adding the following line to the `nagios.cfg` file:

```
broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg
```

The following commands will add that line as well as an extra line that explains what the module is for.

```
sudo sh -c 'printf "\n\n# NDOUtils Broker Module\n" >> /usr/local/nagios/etc/nagios.cfg'
sudo sh -c 'printf "broker_module=/usr/local/nagios/bin/ndomod.o config_file=/usr/local/nagios/etc/ndomod.cfg\n" >> /usr/local/nagios/e
```

## Restart Nagios

Now you need to restart Nagios to use the NDO broker module. Different Linux distributions have different methods of restarting Nagios Core. We will also check the status of the Nag

```
sudo /etc/rc.d/init.d/nagios restart
sudo /etc/rc.d/init.d/nagios status
```

The last command should show Nagios running:

```
nagios (pid 5345) is running...
```

## Check NDOUtils Is Working

There are a couple of different ways to ensure NDO2DB is working.

This command will show Nagios successfully loaded the NDO module:

```
grep ndo /usr/local/nagios/var/nagios.log
```

The last command should output something like:

```
[1475544660] ndomod: NDOMOD 2.1.1 (09-06-2016) Copyright (c) 2009 Nagios Core Development Team and Community Contributors
[1475544660] ndomod: Successfully connected to data sink. 0 queued items to flush.
[1475544660] ndomod registered for process data
[1475544660] ndomod registered for timed event data
[1475544660] ndomod registered for log data'
[1475544660] ndomod registered for system command data'
[1475544660] ndomod registered for event handler data'
[1475544660] ndomod registered for notification data'
[1475544660] ndomod registered for service check data'
[1475544660] ndomod registered for host check data'
[1475544660] ndomod registered for comment data'
[1475544660] ndomod registered for downtime data'
[1475544660] ndomod registered for flapping data'
[1475544660] ndomod registered for program status data'
[1475544660] ndomod registered for host status data'
[1475544660] ndomod registered for service status data'
[1475544660] ndomod registered for adaptive program data'
[1475544660] ndomod registered for adaptive host data'
[1475544660] ndomod registered for adaptive service data'
[1475544660] ndomod registered for external command data'
[1475544660] ndomod registered for aggregated status data'
[1475544660] ndomod registered for retention data'
[1475544660] ndomod registered for contact data'
[1475544660] ndomod registered for contact notification data'
[1475544660] ndomod registered for acknowledgement data'
[1475544660] ndomod registered for state change data'
[1475544660] ndomod registered for contact status data'
[1475544660] ndomod registered for adaptive contact data'
[1475544660] Event broker module '/usr/local/nagios/bin/ndomod.o' initialized successfully.
```

This command will show you the database with populated data:

```
echo 'select * from nagios.nagios_logentries;' | mysql -u ndoutils -p'ndoutils_password'
```

The last command should output something like:

logentry_id	instance_id	logentry_time	entry_time	entry_time_usec	logentry_type	logentry_data	realtime_data	info
1	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868450	262144	ndomod registered for log data'	1 1	
2	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868462	262144	ndomod registered for system command data'	1 1	
3	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868467	262144	ndomod registered for event handler data'	1 1	
4	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868472	262144	ndomod registered for notification data'	1 1	
5	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868479	262144	ndomod registered for service check data'	1 1	
6	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868486	262144	ndomod registered for host check data'	1 1	
7	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868491	262144	ndomod registered for comment data'	11	
8	1	2016-10-04 12:31:00	2016-10-04 12:31:00	868496	262144	ndomod registered for downtime data'11		
9	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869866	262144	ndomod registered for flapping data'11		
10	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869878	262144	ndomod registered for program status data'	1 1	
11	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869884	262144	ndomod registered for host status data'	1 1	
12	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869888	262144	ndomod registered for service status data'	1 1	
13	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869893	262144	ndomod registered for adaptive program data'	1 1	
14	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869897	262144	ndomod registered for adaptive host data'	1 1	
15	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869902	262144	ndomod registered for adaptive service data'	1 1	
16	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869906	262144	ndomod registered for external command data'	1 1	
17	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869911	262144	ndomod registered for aggregated status data'	1 1	
18	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869915	262144	ndomod registered for retention data'1	1	
19	1	2016-10-04 12:31:00	2016-10-04 12:31:00	869920	262144	ndomod registered for contact data'	11	
20	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871043	262144	ndomod registered for contact notification data'	1 1	
21	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871055	262144	ndomod registered for acknowledgement data'	1 1	
22	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871062	262144	ndomod registered for state change data'	1 1	
23	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871067	262144	ndomod registered for contact status data'	1 1	
24	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871072	262144	ndomod registered for adaptive contact data'	1 1	
25	1	2016-10-04 12:31:00	2016-10-04 12:31:00	871077	262144	Event broker module '/usr/local/nagios/bin/ndomod.o'	initia	
26	1	2016-10-04 12:31:00	2016-10-04 12:31:00	874858	262144	Successfully launched command file worker with pid 6026		

## Service Commands

These commands are for starting / stopping / restarting / status ndo2db.

```
sudo launchctl start org.nagios.ndo2db
sudo launchctl stop org.nagios.ndo2db
```

## Service Restart Order

When restarting the MySQL / Mariadb, NDOUtils and Nagios services you should take the following into consideration.

Nagios depends on NDOUtils.

NDOUtils depends on MySQL / Mariadb.

Hence if you wanted to restart the MySQL / Mariadb service it is advisable to stop Nagios first and then NDOUtils. After the MySQL / Mariadb service is up again you can then start NDC

Following this method ensures data is correctly written to the database. You don't specifically have to follow this order, both Nagios and NDOUtils will queue the data until the depend

## Final Thoughts

This completes the installation of NDOUtils and configuring Nagios to use the NDO broker module.

For any support related questions please visit the [Nagios Support Forums](#) at:

<http://support.nagios.com/forum/>

emerge --config =dev-db/mysql-5.6.35

Posted by: **tlea** - Mon, Feb 1, 2016 at 11:06 PM. This article has been viewed 10344 times.

Online URL: <https://support.nagios.com/kb/article/ndoutils-installing-ndoutils-406.html>