



# MIKE Zero 2024 for Linux Installation Guide

## System Requirements

- Intel® 64 architecture processor (x86-64) or compatible non-Intel processor. Please note, that other processor architectures such as Power PC, Itanium, SPARC or ARM are not supported.
- For the best experience, a multi-core or multi-processor system is recommended.
- When utilizing MPI based parallelization in a cluster high-performance interconnection such as InfiniBand is recommended.
- Supported Linux distribution is Red Hat Enterprise Linux 7 and 8 and CentOS 7 and 8

These are the Linux distributions used for building and testing MIKE Zero 2024 for Linux. Other distributions may or may not work and are not recommended. However, MIKE Zero for Linux has been successfully installed on a number of other Linux distributions, including SUSE LINUX Enterprise Server, ScientificLinux and Ubuntu. Please note, that MIKE Zero for Linux is only available for 64 bit Linux distributions.

## Content

MIKE Zero 2024 for Linux consist of the four tar-zipped packages listed below. The tree packages labeled rhel7 and contain the core components, additional tools and prerequisites support both Red Hat/CentOS 7 and 8.

MIKE_Zero_2024.tgz	core components
MIKE_Zero_2024_Tools.tgz	additional tools
MIKE_Zero_2024_Examples.tgz	examples
MIKE_Zero_2024_Prerequisites.tgz	prerequisites

MIKE Zero 2024 for Linux includes the calculation kernels listed below and they are all included in the core component package.

Product name	Application name
MIKE 21 Flow Model FM	FemEngineHD
MIKE 21 Flow Model FM (with GPU support)	FemEngineHDGPU
MIKE 3 Flow Model FM	FemEngineHD
MIKE 3 Flow Model FM (with GPU support)	FemEngineHDGPU
MIKE 3 Wave FM	FemEngineHD
MIKE 21 Spectral Waves FM	FemEngineSW
MIKE 21/3 Coupled Model FM	FemEngine
MIKE 21/3 Coupled Model FM (with GPU support)	FemEngineGPU

The calculation kernels utilizing GPU requires a Nvidia graphics card with compute capability 6.0 or higher. Please note that some of these graphics' cards have varying performance in single compared to double precision calculations. The GPU functionality is based on version 12.2 of the Nvidia® CUDA® Toolkit.



MIKE Zero for Linux utilizes Intel MPI Library and therefore requires an Intel MPI Library runtime environment in order to run the MPI-based applications. MIKE Zero 2024 for Linux has been build and tested with Intel® MPI Library 2021 Update 7 for Linux. The runtime environment packages are included in the prerequisite package.

Other versions of the Intel MPI Library runtime environment may work with MIKE Zero 2024 for Linux, but it is strongly recommended to use the version included in the prerequisite package.

## Installation

To perform an installation MIKE Zero 2024 for Linux you need to unpack the packages and run `install.sh` inside each package directory. The default install location of MIKE Zero 2024 for Linux is `/opt/MIKE/2024`, when installing as root, and `$HOME/MIKE/2024` when installing as a normal user. The MIKE Zero 2024 for Linux packages are installed via the command

```
./install.sh
```

On a cluster with multiple compute nodes MIKE Zero 2024 for Linux and Intel MPI Library must be available on all compute nodes. The standard approach is to install MIKE Zero 2024 for Linux and Intel MPI Library in a location that is shared across all compute nodes. Intel MPI is installed via the command

```
sh /_mpi_oneapi_p_2021.7.0.8711_offline.sh
```

## License

MIKE Zero 2024 for Linux supports both network license and internet license. Dongle based local license is currently not supported. Network license requires a DHI License Manager 2024 installation, which is currently only available for Windows. MIKE Zero 2024 for Linux is as default configured to use network license and license server name can be specified in the installation process.

The license configuration tool named `licconfig`, which is included in MIKE Zero 2024 for Linux, can be used to setup MIKE Zero 2024 for Linux to utilize internet license and to change the default license configuration.

Configuration of internet license is done via the command

```
licconfig set --type=internet --iuser=user@my-company.com --ipassword=my-password
```

Use of Internet license requires applications to communication with an internet license service at <https://dk.license.mikebydhi.com/svc> and <https://license2.mikebydhi.com/svc> via HTTPS (HTTP over TLS/SSL) on port 443, which must be open for outbound traffic.

Applications also start a relay service, when internet license is used. This service uses custom protocol similar to network license on TCP ports 61768 – 61798. The relay service uses the first port in this range that is open for inbound traffic. This inbound traffic is only used by the relay service on the local host to communicate with applications. The use of the relay service can be bypassed by using the license type `internet_direct` instead of `internet`.

Configuration of network license is done via the command

```
licconfig set --type=network --server=myserver.domain
```

Use of network license requires applications to communication with a network license service via custom protocol using RPC (remote procedure call) on port 1799, which must be open for inbound traffic.

More information about `licconfig` can be found via the command

```
licconfig --help
```





## Execution

Before executing simulations, you need to setup the runtime environment for the Intel MPI Library and for MIKE Zero 2024 for Linux. This is done via the below commands

```
source <install path>/intel/oneapi/mpi/2021.7.0/env/vars.sh
```

```
source <install path>/MIKE/2024//mikevars.sh
```

Two examples of how to execute a simulation is listed below. The first is a MIKE 21 Flow Model FM example and the second is a MIKE 21 Spectral Waves FM example.

```
cd <install path>/examples/MIKE_21/FlowModel_FM/HD/Lake  
FemEngineHD lake.m21fm
```

```
cd <install path>/examples/MIKE_21/SW/Island  
FemEngineSW Setup.sw
```

When running FemEngineHD or FemEngineSW as shown above the two computational engines will utilize OpenMP parallelization only. However, it is recommended to utilize MPI parallelization, which is more efficient even for a small number of cores and scalable for large number of cores. To execute the simulations using MPI parallelization you need to use mpirun. To execute the example with 8 cores you need to use the following commands.

```
mpirun -n 8 FemEngineHD lake.m21fm
```

```
mpirun -n 8 FemEngineSW Setup.sw
```

If the number of available cores on the compute node is larger than the number of cores specified in the mpirun command, additional cores is be used for OpenMP parallelization if enough cores are available. OpenMP parallelization can be disabled (limited to only one thread) via the command

```
export OMP_NUM_THREADS=1
```