

# Sharing Open-Source Software at MMSys

The papers accepted to the ODS track that describe new open-source software will need to offer the code publicly. Authors will be permitted (and encouraged) to host the software in their own GitHub repository, but once the camera-ready version of the paper is due, we will create a fork of the authors' repository to the [MMSys GitHub repository](#) to ensure the code represents exactly what was described in the paper. The authors can continue to develop their software even after the camera-ready deadline, but those changes will not be present in the fork. People visiting the MMSys GitHub repository can then choose to download the original code that accompanied the paper or the updated code in the authors' GitHub repository.

Authors should ensure an appropriate license is assigned to the code. GitHub has created [this](#) page that can help choose a good license. If no license is chosen, it will default to the standard copyright laws, which means the software will not be open source and therefore, is not suitable for the ODS track of MMSys. Another useful GitHub page is [this](#) one.

For the reviewers, we want to make life as easy as possible, where we want them to primarily focus on the functionality the software offers rather than how clean/modular/beautiful the underlying code is. We do not want reviewers to have to install dependencies and deal with potential security issues when installing software from possibly untrustworthy sources. Therefore, the authors will need to offer a 'ready-to-go' bundle, so the reviewers can instantly start using the software (unless the source code really is self-contained and absolutely trivial to compile). Depending on the type of software, there are different options we suggest for authors to create such a bundle:

- Host the software on a web server, so that reviewers only need a browser to access the full functionality of the software. This option is generally only suitable for lightweight applications and when the web server is guaranteed to be responsive. Authors should be vigilant that the software remains accessible, as a broken or bad experience by the reviewers will likely lead them to reject the paper.
- Use one of the tools that allow direct integration of your artifacts into the ACM DL: these tools (Collective Knowledge, OCCAM and Code Ocean) are each described with short videos [here](#). These cover a wide range of cases and programming languages, and are worth considering in most cases. Note that the code must be accessible and easily runnable, not presented as a blackbox.
- Host the software in a virtual machine, so that reviewers only need to load the disk image. Using a virtual machine ensures all dependencies have been installed and that reviewers can directly start using the software.

The last option may seem daunting, but it is actually quite simple. Here are some instructions for setting up virtual machines that use Windows and Linux as operating systems. To get VirtualBox, the virtual machine manager, go to <https://www.virtualbox.org/> and download/install the latest version.

## Windows

Microsoft offers Windows versions free of charge for use in virtual machines. The disk images expire after 90 days, which is more than sufficient to cover the reviewing period.

- Navigate to <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/> and download one of the disk images that best suits the conditions of the software to install, preferably Windows 10. Choose VirtualBox as the platform.
- Unzip the archive, which produces a file with the .ova or .ovf extension.
- Launch VirtualBox, choose File → Import Appliance and select the .ova file.
- Click on the imported virtual machine, choose Settings and make changes if necessary, such as the amount of memory to allocate, the number of processors the virtual machine has access to, USB compatibility mode, etc. Remember that reviewers may not have a powerful machine, so only set the options at the lowest acceptable value for the software to properly run.
- Click the Start button to launch the virtual machine.
- Once the virtual machine is up and running, install the latest device drivers/emulators by choosing Devices → Insert Guest Additions CD Image, and Windows should then automatically offer to run the VBoxWindowsAdditions program, which is what we want.
- After rebooting, install any latest software updates if needed, but in general it is now ready for use.
- Install the open-source software and all dependencies.
- To submit the software for review, the virtual machine needs to be exported. First, shut down the OS, and then in the main VirtualBox application choose File → Export Appliance. Store it with a memorable name, and keep all settings at their defaults.
- Note that if more than a couple of days or weeks have passed since installing the OS, be sure to repeat the entire procedure first before exporting the virtual machine with the software for review, due to the 90-day expiration period. Be aware that most reviewers will likely look at the submission in the last week before the review deadline. It would be a pity for a reviewer to be confronted with a virtual machine that has expired. Caveats: \* Disable defragmentation in Windows if you are using an SSD hard drive as otherwise it will wear it out. To disable it, click the Start button and enter 'dfrgui' in the command box; in the window that pops up, click 'Configure Schedule', untick 'Run on a schedule', and click the OK button to confirm. \* Always shut Windows down cleanly by going to Start → Shut down, rather than by using the 'X' button of the VirtualBox menu bar.

## Linux

Here, we walk through setting up Ubuntu, but installing other Linux flavors follows similar steps.

- Navigate to <https://www.ubuntu.com/download/desktop> and download the latest version.
- Unlike the Windows image that was already prepared as a virtual machine by Microsoft, we still need to install the Ubuntu operating system first.
- Launch VirtualBox and click the 'New' button. Give it a name (e.g., Ubuntu), set the type to 'Linux' and the version to 'Ubuntu (64-bit)'. Click continue.
- Set the desired properties of the operating system in terms of memory and hard disk. You can try working with these defaults and change the settings later if needed.

- Before powering on the machine, we need to attach the Ubuntu installer as a virtual CD-ROM drive. Click the 'Settings' button and go to the 'Storage' tab. Select the 'Controller: IDE' from the list and click the little icon that looks like a CD-ROM with a plus sign. Choose the .iso file you downloaded and click 'OK'.
- Click the 'Start' button to launch the virtual machine. The Ubuntu installer should activate. Click 'Install Ubuntu' to begin, and select the two checkboxes to download updates and install supporting software. Give the machine a descriptive name and choose a reasonable username and a simple password (e.g., 'mmsys'). Tick the checkbox to automatically login upon startup so reviewers will not have to type in the password, but you will need to share it with them anyway just in case.
- Once the installation is finished and it asks you to restart, instead shut it down by clicking the 'X' in the corner of the window and selecting 'Send the shutdown signal'. Then, go back to the Settings and remove the CD-ROM image if it is still there.
- Click the Start button to launch the virtual machine.
- Install essential tools by opening the Terminal and executing 'sudo apt-get install build-essential', which will prompt you for the password you set.
- Finally, follow the same steps as for Windows as listed above to install the additional software and for instructions on how to export the virtual machine image for review. As Ubuntu does not expire after 90 days, you do not have to worry about the expiration of the VM.