**ARC System Setup Guide**

**(ESAL Group)**

**Getting started with ARC**

To use the ARC service, you must apply for an individual user account, which must be part of an ARC project.

**Requesting a user account**

You can directly apply for a user account using the [user registration form](https://www.arc.ox.ac.uk/arc-user-registration-page) (<https://www.arc.ox.ac.uk/arc-user-registration-page>).

Once the application is approved, you will receive an email containing your ARC login credentials: a username and a temporary password. The email also contains instructions how to change your temporary password.

**Access to ARC**

Access is available only from within the University of Oxford network. If you are not on the University network, you should use the University VPN service to connect. If you are unable to use the VPN service, you may be able to register your static IP with the ARC team (support@arc.ox.ac.uk) to enable access.

**Connecting from Windows**

You need to download and install an application that allows Windows to connect - a popular example is MobaXterm. MobaXterm for Windows can be found at <https://mobaxterm.mobatek.net/>

**MobaxTerm interface**

In the MobaxTerm interface, click on Session (the icon on the top left corner), fill in the “remote host” with **gateway.arc.ox.ac.uk,** and fill in the “Specify username” box with your username (Oxford SSO like “engs####”).

A screenshot of a computer

Description automatically generated

Once the session is created, in the next connections, all you need to do when you open the MobaxTerm interface is to select the session and login with your university username when asked.

A screenshot of a computer

Description automatically generated

**A screenshot of a computer screen

Description automatically generated**

Then with asking your password, type it again.

You are now in the ARC $HOME. Do not transfer data into the home directory on gateway.arc.ox.ac.uk and use your $DATA path “under /data/engs-psal/” instead as following picture.

A screenshot of a computer

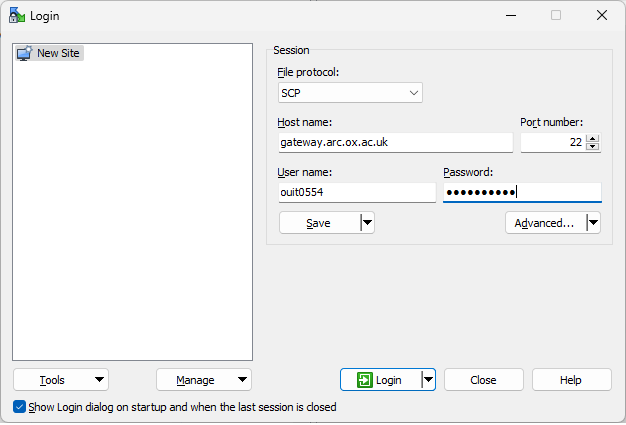
Description automatically generated

**Copying to/from ARC using Graphical File Transfer utilities**

You can use tools such as **WinSCP** utility to copy files.

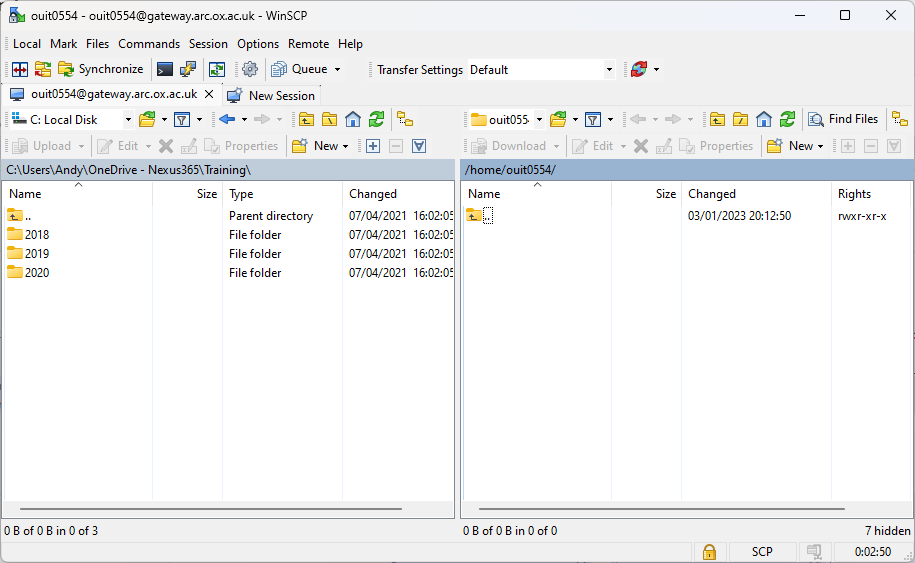
First open **WinSCP** and complete the Session fields as follows:

**File Protocol:** SCP **Host Name:** gateway.arc.ox.ac.uk **Port Number:** 22 **User Name:** Your ARC username **Password:** Your ARC password



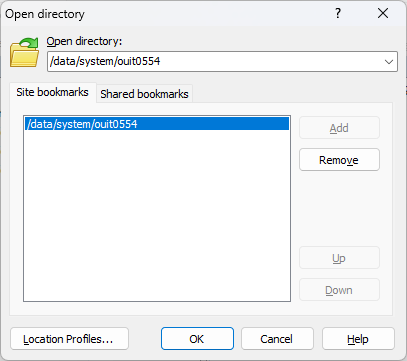
If you wish, you can click Save to save this information for future sessions, otherwise click Login to connect…

You should then be logged in and see the following type of display…



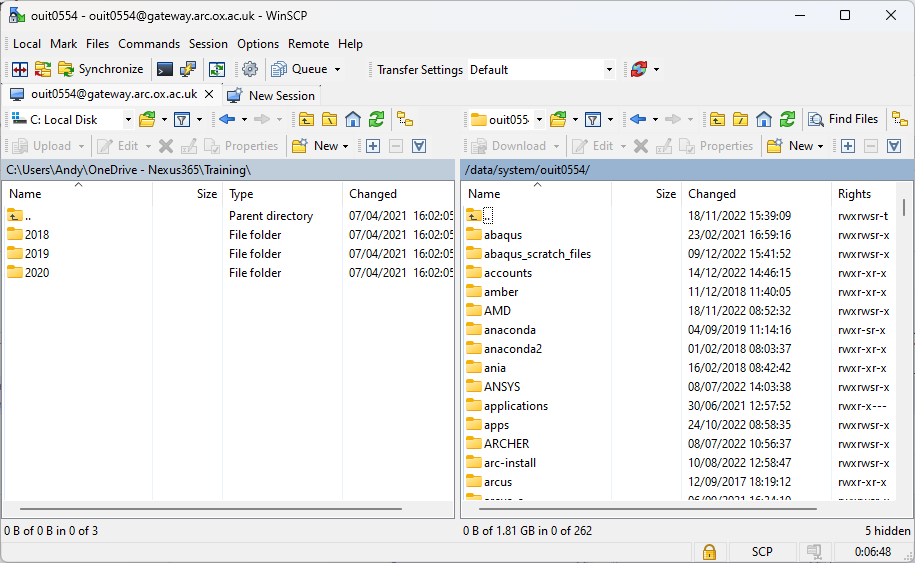
The left “pane” is the local filesystem on your computer, and the right “pane” is the remote ARC filesystem. This defaults to your home directory on gateway.arc.ox.ac.uk which is not the same place as $HOME on ARC - so do not transfer files here.

Use the WinSCP menu options, Remote | Go To | Open Directory/Bookmark… to bring up the following dialogue box:



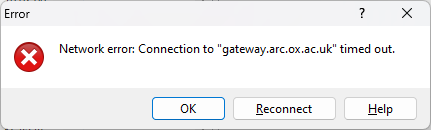
In the **Open Directory** selector, type in the path to your $DATA area. For example /data/system/ouit0554

You can click Add to save this as a bookmark for next time, or simply click OK to open this directory on ARC.

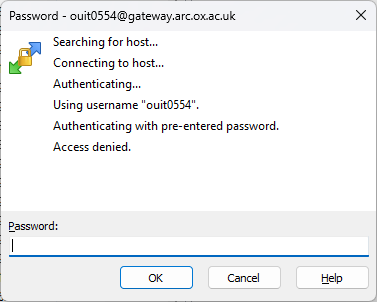


You should now see your $DATA area on the right pane and you can drag/drop files between your local and ARC filesystems.

If you see an error of this type:



This indicates a network problem between your local computer and the ARC service. Typically, this is caused by being off campus **without** a working university VPN connection. Try restarting the VPN client. If you supply the wrong username or password to the file transfer utility, you may see errors such as the following:



**Access Denied** commonly means that you have made a connection to ARC, but you have supplied the wrong username/password combination. Try checking these using a standard SSH connection.

**Creating your own virtual environment**

You must create your conda environments from a SLURM interactive session. So, ensure you have an active interactive session by running:

srun -p interactive --pty /bin/bash

You should decide which version of Python you wish to use, 2 or 3. There are Anaconda modules available for both versions, the current Anaconda versions can be found by typing:

module spider anaconda

To load the version of Anaconda you want, use one of the following commands:

Python 2:

module load Anaconda2

Python 3:

module load Anaconda3

or one of the specific Anaconda versions shown by module spider.

Once the module is loaded, you can use the conda commands to create a virtual environment in your $DATA area. For example, to create an environment named myenv in $DATA, you can use the following commands:

export CONPREFIX=$DATA/myenv

Python 2:

conda create --prefix $CONPREFIX --copy python=2.7

Python 3:

conda create --prefix $CONPREFIX

**Submission Script**

First we use the nano editor to create the file:

nano submit.sh

This command will start the Linux nano editor. You can use nano to add the following lines:

#! /bin/bash

#SBATCH --nodes=2

#SBATCH --ntasks-per-node=4

#SBATCH --time=00:10:00

#SBATCH --job-name=myjob

#SBATCH --partition=medium

module load Anaconda3

python ./Job01.py

The first line #! /bin/bash tells Linux that this file is a script which can be run by the BASH shell interpreter.

The following #SBATCH lines request specific cluster resources:

--nodes=2 requests two ARC nodes

--ntasks-per-node=4 requests 4 cores per node (a total of 8)

--time=00:10:00 requests a run time of 10 minutes (the maximum for the devel partition)

--job-name=myjob request your set job

--partition= medium requests that this job runs on the medium partition

Note: Clusters have the following time-based scheduling partitions available:

* short (default run time 1hr, maximum run time 12hrs)
* medium (default run time 12hrs, maximum run time 48hrs)
* long (default run time 24hrs, no run time limit)
* devel (maximum run time 10 minutes - for batch job testing only)
* interactive (maximum run time 24hrs, can oversubscribe, for pre/post-processing and building software)

module load Anaconda3 The module load command is used to make an application environment available to use in your job, in this case the Anaconda3 application.

python ./Job01.py Run the python code file

**Remark:** if you're getting the "error: unknown option script" error, it's because you've edited the job script on a Windows machine (using MobaXterm's view/edit feature or windows Notepad) and it needs to be converted to Unix format.

You can do this by running the below line in MobaXterm:

* dos2unix <your\_script>.sh

**Connecting from Linux**

Linux and Mac users should use ssh from a terminal to connect to the ARC systems. To connect to the ARC cluster use: ssh -X username@arc-login.arc.ox.ac.uk

If you are connecting from outside the University network (or from the VPN service) you will need to connect via gateway.arc.ox.ac.uk - ensure you specify your ARC username in the ssh command, as shown below:

ssh -X username@gateway.arc.ox.ac.uk

You can then SSH to arc-login as required.