

# Cache files

Nov 2023

Michael Benjamin, mikerb@mit.edu  
Department of Mechanical Engineering  
MIT, Cambridge MA 02139  
project-pavlab/mtasc/mt\_cache\_files

---

<b>1</b>	<b>Cache Files</b>	<b>1</b>
1.1	Cache Files Example Snippets . . . . .	2
1.2	SSH Keys . . . . .	2
1.3	Generating the Cache Files . . . . .	2
1.4	Aliases . . . . .	3

---

## 1 Cache Files

The MTASC network environment, in terms of which machines are powered and with which IP address, is somewhat non-deterministic. This is due to intermittent failures of machines and network IP assignments not being guaranteed. This may vary from system to system or day to day. However, since the pablo network configuration is fairly stable during a single power cycle of the system, the configuration is cached in four files in the user's home directory.

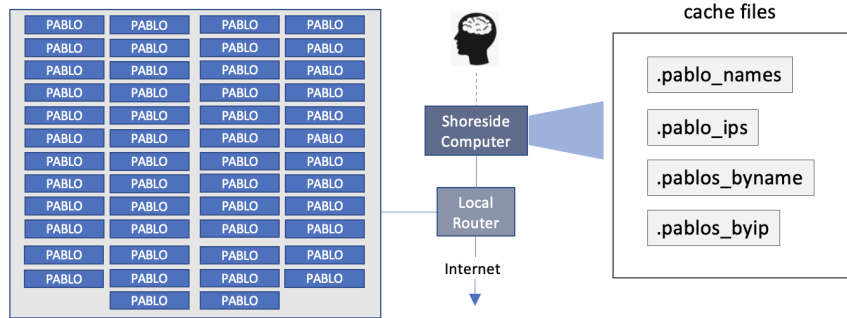


Figure 1: **Cache Files:** The cache files are stored in the home directory of the user operating the shoreside command-and-control computer.

These files are used by various launch scripts to allocate PABLO computers for any given multi-vehicle launch. See [?].

- `.pablo_ips`: List of PABLO IP addresses on the local network.
- `.pablo_names`: List of names for each PABLO on the local network.
- `.pablos_byname`: Mapping from PABLO name to IP address.
- `.pablos_byip`: Mapping from IP address to PABLO name.

## 1.1 Cache Files Example Snippets

Example `/.pablo.names:`

```
paba01  
paba02  
paba03  
paba04  
paba05
```

Example `/.pablo.ips:`

```
192.168.7.32  
192.168.7.33  
192.168.7.34  
192.168.7.35  
192.168.7.36
```

Example `/.pablos.byname:`

```
paba01 192.168.7.60  
paba02 192.168.7.66  
paba03 192.168.7.71  
paba04 192.168.7.62  
paba05 192.168.7.55
```

Example `/.pablos.byip:`

```
192.168.7.32 paba25  
192.168.7.33 paba24  
192.168.7.34 paba21  
192.168.7.35 paba40  
192.168.7.36 paba22
```

## 1.2 SSH Keys

Generation of cache files, and most other interactions with the PABLO cluster, will require the user to have an ssh-key installed on the local laptop and credentialed with the PABLOS. Otherwise remotely executed steps under the hood of bash scripts will likely fail.

See the document on MTASC SSH Keys.

## 1.3 Generating the Cache Files

(Re)generating the cache files is recommended as a first step each time the MTASC system is powered on.

The cache files are generated by a script, `find_pablos.init.sh` which can be found:

```
moos-ivp-swarm/scripts/find_pablo_init.sh
```

This script will scan a range of IP addresses using the `fping` utility to determine which IP address seem to have a live computer. It will then attempt to run a remote command on each computer to get the name of the PABLO. For machines that are not a PABLO, this command will return an empty string. If a machine is indeed a PABLO, a new entry is made for each of the four cache files.

## 1.4 Aliases

Below are a list of aliases that are recommended to be part of your bash environment of your local computer being used to control an MTASC cluster.

```
pup='find_pablo_init.sh -v'

PABLO_FILES=" ${HOME}/.pablo_names"
PABLO_FILES+=" ${HOME}/.pablo_ips"
PABLO_FILES+=" ${HOME}/.pablos_byname"
PABLO_FILES+=" ${HOME}/.pablos_byip"
alias hhp='head -n 50 ${PABLO_FILES}'
alias hp='head -n 5 ${PABLO_FILES}'
```

A typical initial action upon starting the MTASC cluster, and letting them boot is to run:

```
$ pup
```

Then to verify things work, taking a look at say the first 5 lines of each file:

```
$ hp
```

Or `hpp` if you want to see the full contents (up to 50 lines) of each file.