

Getting Started with Tiny Core Linux

Tiny Core Linux, not being a complete Desktop, requires the internet to access to our repository of applications. Everyone should take the time to read this document and understand the design philosophy behind Tiny Core Linux. This will insure a more pleasant experience.

Four Modes of Operation

The First Mode: Cloud/Internet

The first mode of operation is the default boot mode of Tiny Core Linux. It is what I call the Cloud Mode, or the Internet Mode. Tiny Core boots entirely into RAM. Then using the internet and the Apps icon, a shortcut to our appbrowser GUI, one can begin to explore our application extension repository. By the way, there are two very different kinds of extensions available, more on that later. In this mode of operation, extensions downloaded via the appbrowser will be for this session only. That is, the extension will download and install into RAM. The downloaded extension is not saved. This is to maximize the use of available RAM. This is the most nomadic mode of operation but it also requires the most resources.

The other modes of operation is by using persistency. Booting with certain options one can persist downloaded extensions.

The Second Mode of Operation: PPR/TCE

The second mode of operation is the use of the boot option `tce=hdXY`, i.e., specifying a writable persistent storage partition and using our TCE repository of applications. This will become your Persistent Personal Repository (PPR). Upon further booting all TCE extensions will automatically be loaded into RAM. Any further downloading of TCE type extensions from the appbrowser (Apps) will persist in this specified location. Note the default directory name is "tce". However this can be user specified, e.g., `tce=hdXY/mystuff`. Using this second method of operation, PPR, is preferred as it always results in a known pristine state of your operating environment. TCE type extensions are basically a `tar.gz` archive with optional included menu and/or icon. These extensions reside in RAM when used in the PPR mode of operation. These would be called WORMS, the good kind! That is Write Once Read Many. Adding many TCEs may quickly exhaust your systems memory, which can become problematic for users with low RAM.

Once a PPR has been setup via a one time only boot code of `tce=hdXY` all subsequent booting will load, install, or mount your PPR collection. Even this boot code can be dropped. If no boot option specifying the tce location is used then the system will autoscan for the default tce directory. Recall, I mentioned that you can also specify the directory name. If you have chosen to use a different name via boot option, e.g., `tce=hdXY/mystuff` then you must continue to specify that boot code. The system will only scan for the default "tce" directory. The auto-scan was designed to have a "hands free" boot from a read-only boot media, i.e., the downloaded `tinycore.iso` written to `cdrom`. However this is a benefit to always specifying your PPR location even when using the default tce. By specifying the location, the system does not have to autoscan. This saves time during booting.

The Third Mode of Operation: PPR/TCZ

When using a PPR one has the choice of TCE or TCZ extension types. To maximize memory usage Tiny Core offers the TCZ extension type. These extensions are mounted from your PPR and therefore use significantly less RAM. TCZ extensions are cramfs or ziofs compressed mountable images of an application directory, which become symlinked into the root filesystem. You may mix and match either extension type. Note some extensions may not be available as a TCZ. Using TCZ mounts will provide the maximum number of extensions for any given RAM size. All other aspects of using a PPR, i.e., initial setup and boot type options are the same as the above mentioned PPR/TCE. Using the boot code of base will ignore any PPR.

The Fourth Mode of Operation: PPI/TCE

The fourth mode of operation literally installs extensions into a Linux partition or a loop back file. This provides a hybrid type of installation, where core is in RAM and the extensions are installed onto a persistent storage device. I call this mode of operation of Persistent Personal Installation or PPI. The benefits are speed, as no loading or mounting occurs during boot. The same RAM savings offered by the mounted PPR/TCZ extensions is another benefit of PPI. However a possible drawback of this is losing the "pristine state" offered by using PPR. Installing applications, as done in most typical environments, can result in "system rot", i.e., over time, or via user/system error, corruption can occur and might not be readily noticeable. By choosing this method one should only use our TCE collection of extensions. The TCE collection being tar gzipped will install into the specified location. To initially setup your PPI use the boot code of `tclocal=hdXY`. This will look for a directory named `tclocal` or a single loop back file of the same name. Note there is a menu option to create a loop back file for this very purpose. Using a PPI you don't need to keep the downloaded TCEs. No need to specify `tce=hdXY` as used in the PPR mode. For a "hands free" boot even the `tclocal=hdXY` can be omitted. But by always specifying the `tclocal` boot code will skip the autoscan and result in faster booting. Using the `nolocal` boot code will always ignore an existing PPI.

With either a PPR or PPI one can maintain their collection of extensions on a pendrive, compact flash, or other removable media for maximum portability.

New Download Capabilities of Tiny Core:

Many improvements were made to enhance the installation of extensions to Tiny Core be it a PPR or a PPI mode of operation. Additional extension codes of "l" or "m" or both will automatically load and configure libraries and modules respectively. These functions occur without any user intervention or configuration. Additionally Tiny Core supports installation scripts. These run upon loading/mounting. The ".dep" extension brings simple download dependency resolution to Tiny Core. This greatly simplifies a successful download installation, loading, or mounting of any application extension. The user does not need to read the ".info" file for the purpose of trying to manually satisfy the application extension run-time dependency requirements. Simply choose the typical application name and all other run-time requirements will also be downloaded and prepared for successful application launch.

Backup/Restore and Other Persistency Options:

Backup/Restore

After setting up your new system, you will want to save your settings. There is a file called `/opt/.filetool.lst`. It is a simple text file of files and directories listed one per line using a full path that the user wants to save/restore. Use any of the GUI tools provided on the Tools menu or any editors available via extensions or vi to change/update (e.g., select Beaver, then open `/opt/.filetool.lst`).

The default is to backup the entire `home/tc` directory. However, one can choose to selectively backup only certain files. However it is important to note that the entry `/opt/.filetool.lst` must NOT be removed from the `.filetool.lst` as this provides for persistence for the `.filetool.lst` itself.

Also note that the capability exists to exclude files from the backup with the file `/opt/.xfiletool.lst`. Adding entries to this file will exclude them from the backup. Cache and other files are in the default `/opt/.xfiletool.lst`.

The backup will be written to a file called "mydata.tgz". You can initially select the storage device by using the boot option of: `restore=hdXY` or after boot, by selecting "Backup/Restore" from the "Panel". Your backup will be automatically searched for and restored during subsequent booting. Once a `mydata.tgz` has been successfully created, the boot option may be omitted. Autoscan once again is there to support "hands free" booting. And as before, always specifying the boot option speeds your boot time.

Additionally you may add a directory to the location of the backup, e.g., `restore=hdXY/a_directory`. This will allow the backup file "mydata.tgz" to be placed in a directory. Using a directory also means that you must always specify it via its boot code. You can skip using your backup with the boot option of `norestore`.

These options together with a local `.xsession` provide the user with much more control of their preferred environment. For example, by editing the `.xsession` you can start up your favorite X-Window programs. You can start non-Window programs by editing the `.profile` and then adding it to your `/opt/.filetool.lst`. You can even load additional modules and system-specific required files by editing the `/opt/bootlocal.sh` file and then adding it to your `filetool.lst`.

Persistent Home

Just as Tiny Core offers persistency options for downloaded application extensions, so does it for your home directory. Use the boot option of `home=hdXY`. The very first use of this code will automatically setup `/home/tc` to "bind" to `/mnt/hdXY/tchome`. By using this option, you are not having to wait for a backup/restore for those items in your `/home/tc` directory. Backup and restore should, as with any system, still be done. The boot option must always be used, as there is no autoscaning available for persistent home. Note again, the initial use of this option will automatically setup your `tc` home account so care should be taken in its use.

Encrypted Home

Persistent home with encryption is also available. However, this is a little different, as we use a loop back file. Tiny Core offers a menu item to initially setup a encrypted loop back file. See the "Tools" menu section for "Make Crypto Home". Once created you will need to choose a password and use it upon every boot. Once successfully created use the boot option of `cryptohome=hdXY` and during the boot process you will once again be prompted to enter your password.