

## DC++ : Using DC++ : How do I resume my downloads in DC++?

**Question:**

**Answer:**

DC++ will by default, automatically search for alternative sources for files in your download queue. If it does not, make sure the option is enabled (File | Settings | Advanced | *Automatically search for alternative download locations*). At the time of this writing, it searches based on the existing filename and the file size. Any matches that are the exact same size are added as alternative sources. The file(s) will now resume downloading. In the near future, DC++ will search for files based solely on its *TTH value*.

You can manually add files into the download queue by similar methods. Search for a file (its best to also specify a minimum file size to reduce the number of unneeded results). If the file is exactly the same size as the one in your download queue, you can right-click on it and select *Download to..* The existing file should be in the submenu. If not, then its not the same size or TTH value.

If the file refused to resume, then you likely have a *Rollback Inconsistency*.

## DC++ : Using DC++ : What is TTH (Tiger Tree Hashing)?

### Question:

### Answer:

Tiger Tree Hashing or TTH is used to verify the integrity of large chunks of data. The data is split into small pieces which are individually hashed, then hashed together until one, root hash is formed. A TTH has looks something like: `4NIABZVGR4HOTNRPOMH2IRIJQSJYKFHEEIQUJWY`. Detailed information can be found [here](#). DC++ 0.400 introduced mandatory TTH. Once DC++ hashes all of your share (yes this **will** take a while) it will only hash new files. The hashing thread in DC++ is set to low priority, so it shouldn't interfere too badly.

There are several benefits of file hashing:

1. No longer does one need to pay attention to the name of the file when looking for alternative sources. If the files are the same, they will have the same hash and thusly be chosen as an alternative source.
2. Mutli-source downloading. While it is currently not implemented, now there is a safe way to implement downloading files from multiple sources. All clients at this point have been implementing multi-source downloading in cowboy fashion. They do **not** verify the files are the same (except for the size and partial name) which does result in corrupt files. A file hash ensures the files are identical.
3. [Magnet Links](#). Once again, it is currently not implemented but it will allow users to initiate downloads from a link on a website similar to other direct download links like an ed2kfile.

## DC++ : Using DC++ : What is a Rollback Inconsistency?

**Question:**

**Answer:**

This message means that somehow the data has become corrupted during a transfer. Usually its because one of the users, either you or the one you are downloading from, is using a bad firewall such as ZoneAlarm which is intercepting some packets. There are three things you can try to fix the problem:

1. You can attempt to fix the file by chopping off the rollback bytes, using a tool like `cutoff`.
2. Delete the file and redownload it
3. Completely disable the rollback feature by changing the value in Settings -> Advanced to 0. You can do this, but at **your own risk**. The rollback is a scheme to ensure that the file you're resuming and the file you're downloading from a remote source are the same. If you disable it after getting the rollback inconsistency message, there **will** be corruption.