



BTDeploy

Paper outlining the BTDeploy program,
and basic setup.

This document provides a basic how-to for a reasonably technically proficient user to create a system image compatible with BTDeploy, and how to actually deploy machines using the software.



Table of Contents

Introduction	2
Prerequisites	2
Preparing your Images.....	2
Generating your Torrent Files.....	4
Actually Building BTDeploy	5
Setting up the server-side.....	6
Settings.ini	7

Introduction

Before you start diving in, I'll explain some basic information about the program. It's simply a GUI that simplifies a lot of other tasks. It isn't some "all in one" solution like Symantec Ghost. It utilises certain programs to achieve certain tasks. Be wary of this when deploying, and also be aware that this program is unsupported other than this document, and email. Note that a reply cannot be guaranteed, but I'll endeavour to reply 99% of the time within 24 hours. **Any sections in this documentation that are in italics are for advanced users. It is not necessary to read these unless you want to pick up some handy tips.**

Prerequisites

Before you get started the following software will be needed to keep upto date, build, and deploy BTDeploy.

- Git – or TortoiseGit [I use both]
- Windows Deployment Services, or equivalent to provide network PXE boot.
- WAIK 3.0 [Windows Automated Installation Kit]. Download from Microsoft.com.
- Symantec Ghost. (optional)
- ImageX [part of WAIK]
- Other programs listed in the .gitignore file in the root directory.
- Drivers for your OS [be it Win7 or Vista]
- uTorrent, or another BitTorrent client with a built in tracker [Azeureus/Vuze maybe?]. If using uTorrent, get the latest beta version rather than the stable. It has bugs if running on Win7 and tends to hang.
- AutoIT3/SciTE4AutoIT3

Preparing your Images

For your images to be compatible with BTDeploy, they don't have to meet a specific format (previously they did – that has been removed with a new revision). Now, you are able to take an 'out-of-the-box' DETA workstation image and deploy it. Actually, you're able to take **ANY** standard WIM image and deploy it. The assumption that BTDeploy makes is that:

- The first image in the wim (WIM's can store multiple images) is for the "System" partition.
- Second image is for the "Data" partition.

This behaviour can be changed with the "Settings.ini" file later on, using options in the "ImageX" section. As a matter of fact, if you were really creative, you could put all 3 images in one wim. So, you could have a wim like this:

1. CFS System Partition
2. CFS Data
3. Desktop System
4. Desktop Data
5. CFT System Partition
6. CFT Data

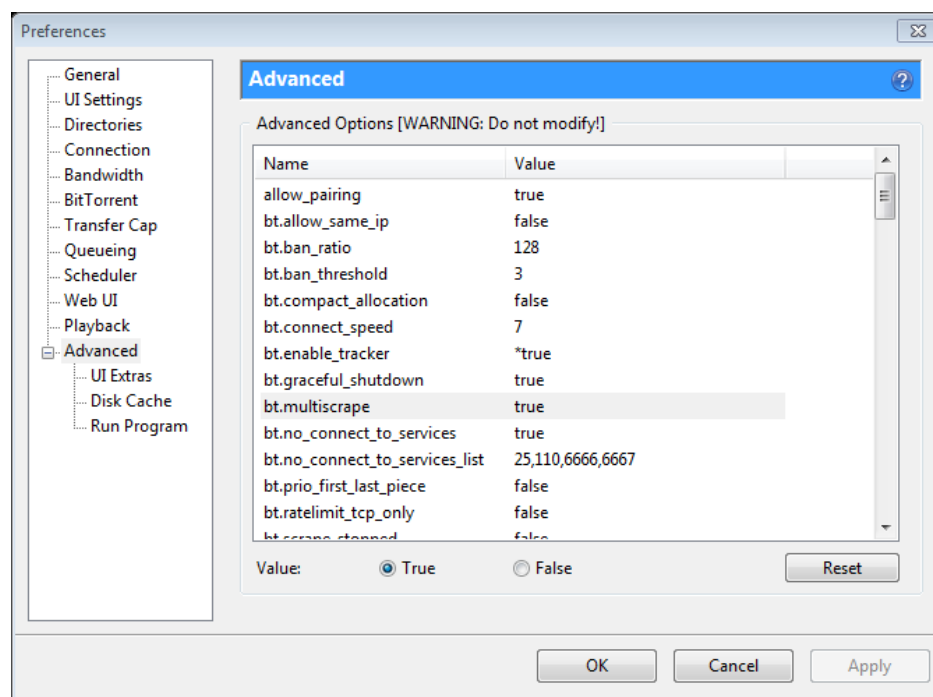
Advantage of doing this, is that by nature, WIM's only store **ONE** copy of each file. Since most of the images are the same (being the same OS, and having the same software), you'll have a very small image. To put it into practical terms, when I had my dual boot CFS image, with a MOE install and a Home install, in a ghost image, it was about 40gb. When I converted this to a WIM (it took ages – basically applying it to a drive, then capturing to wim) – I got it to about 18gb. That's with all of my schools software on it, like Win7, Adobe CS3, ArcGIS, AutoCAD, and a bunch of other large software suites. It's very feasible that a school that combined all 3 images could get their 'master image' to about 30gb (assuming they had **heaps** of software).

Generating your Torrent Files

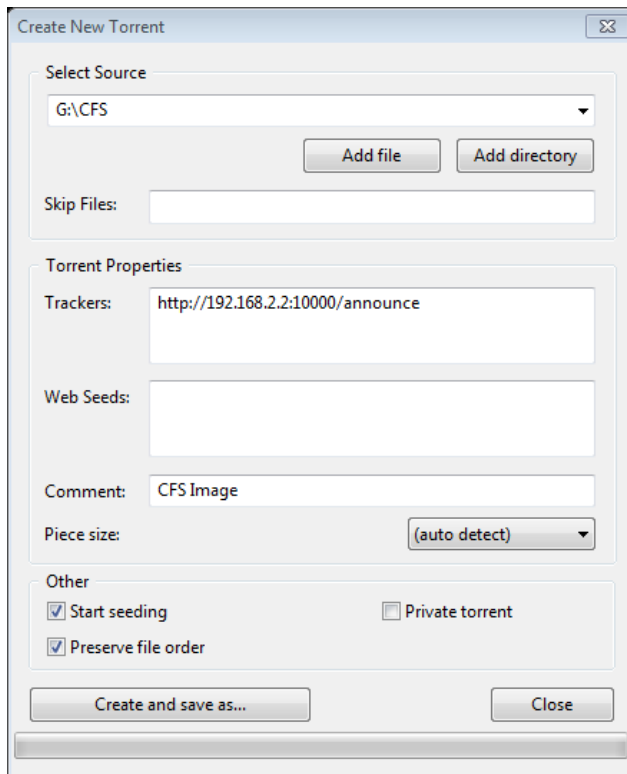
First thing we need to do is generate our torrent files for the install, so BTDeploy has something to actually use. In uTorrent, this is a fairly straightforward setup, and I'm going to assume we have a folder layout something similar to the following for the next step:

- ❖ Root
 - CFS
 - Image.WIM
 - Desktop
 - Image.WIM
 - CFT
 - Image.WIM

Before we can actually generate, we will need to enable tracker support in uTorrent. This can be done by going to the Settings dialog, then clicking on Advanced, and changing '*bt.enable_tracker*' to 'True'.



Once that is done, we will actually generate our torrent. You will need the port number that you setup for uTorrent to run on [**I always choose '10000' – nice round number, easy to remember – it's what I'll use in all the samples from here on out**]. Go to File -> Create new Torrent, and set it up so it's something like the following screenshot:



You can also specify a web seed if you want. A web seed is basically a location on a webserver where the file that you're seeding resides. So, you can place your CFS/Desktop/CFT folders on your IIS server, put the paths to the files in there, and it *should* enable a faster download. I think aria2c actually uses web seeds as a last resort in case there are no regular seeds (don't quote me).

Once you've setup the basics, press "Create and Save as". Ensure the Private Torrent box is unticked, because that will disable DHT and local peer discovery, which can be handy in the instance that your tracker is down, and you want your clients to operate independently.

Actually Building BTDeploy

Here's the hard part – actually building BTDeploy. All of the files you need should be in .gitignore. Not all of them are used anymore [like, MBRWizard/MBRFix]. The main files you need are ImageX, and Ghost (ghost is optional). Assuming you've checked out the git repo exactly as-is, you'll need to do the following:

- Put Ghost, and GDisk32 in the program files folder.
- ImageX in the Windows/System32 folder [whatever it is in the .gitignore].
- Explorer++ in the Program Files folder as described in gitignore.

Once that prep is done, you can move on to editing the ImagePE.bat. At the top of that file there are two variables that describe the locations for your build. Change them to suit. The WPEDIR variable gets "-x86" and "-x64" appended to it, so don't worry about adding those to the end of the path. This is for when there is a 64bit compatible version of BTDeploy out.

Now, the next modification. Open up AyrSHSDeploy.au3 in SciTE4AutoIT3, and look at the top line. Modify that to the path of your BTDeploy "Program Files" folder, otherwise

updates/recompiles won't go to the right place. Once that is done, build the executable. It should end up in the ImagePE directory.

Now, the next modification is to configure the deployment path [where BTDeploy will fetch your torrents from]. This is a reasonably simple one – go to the path where you put BTDeploy and go inside the “Program Files\DETA” folder, and you should see a Settings.ini. Change the relevant setting in there.

Now, go to the start menu and start an elevated “Deployment Tools Command Prompt”. You can do this by searching for ‘deploy’ in the start menu, then right clicking on the result, and click ‘Run as Administrator’. This is needed because of the driver integration stuff that BTDeploy does. Just say, for example, that you put your BTDeploy root folder in R:\Work\BTDeploy, type the following commands into the prompt:

```
R:
cd Work\BTDeploy
ImagePE.bat
```

Answer ‘y’ to removal of the directories. Once the iso is built, you should be able to test it in VMWare or VirtualBox with no troubles. In my experience, VirtualBox seems much faster at booting WinPE, so use that to test it with your images to prove that everything is working correctly before you push it into production. **Always assume that code in the BTDeploy repository is untested. Testing usually occurs after each commit.**

Setting up the server-side

So, one major thing you should know is to always make sure that your deployment box has a static IP, otherwise you'll have to regenerate the torrents [to change the tracker] each time the DHCP lease on that workstation expires.

Basically, put your torrents and your disk wipe scripts where you set your deployment path. Something like this:

- ❖ Root
 - CFS.torrent
 - Desktop.torrent
 - CFT.torrent
 - CFT (folder)
 - Image.WIM
 - Disk.wipe.txt
 - CFS (folder)
 - Image.WIM
 - Disk.wipe.txt
 - Desktop (folder)
 - Image.WIM
 - Disk.wipe.txt

Settings.ini

Put simply, most settings in the “Settings.ini” are fine at their defaults. Only ones most people should be concerned about are in the “[Main]” section. The “ImageX” section deals with the indice in the WIM that the specific partitions image is located at. For example, a standard config will have the System partition set to 1 and Data set to 2 (doesn’t matter if the “Data” partition isn’t present in the WIM though – BTDeploy checks for this).

Problems?

Okay – heres where it gets interesting. Inevitably you’re going to run into problems. Here are the most common problems/solutions.

1) I get a “something went wrong” error in the debug box at the end when I try to apply an image. Whats wrong?

This error could mean a lot of things. Main causes:

- *Is your “Data” partition big enough for the WIM? BTDeploy downloads it to there, so if its not big enough – then it freaks out when trying to download.*
- *Did you cancel halfway through? If you cancel an image applying, its best to restart the PE environment from scratch. Sometimes old disk.wipe.txt scripts are left over.*
- *Is your deployment path correct, and can BTDeploy access it? Check in a web browser. Also check if your IIS server has mime types configured for .torrent. By default IIS is configured to **NOT** serve files it doesn’t have a MIME type for.*
- *Is your Deployment path setup correctly? Check the file paths. Check if the torrents work by downloading them in another client.*

2) Downloads take forever, or they stall.

Okay, one reason for downloads taking forever – there are no seeds. Check if your seeds are connectable (uTorrent shows clients that are connecting to the tracker). Could also fix this by adding web seeds.

There is a downside to web seeds though. If there are no seeds, and you start a lab of 30 machines – they’ll try to grab it from your web seeds (intended behaviour), but, that one machine obviously isn’t on a 10 gig link – so its essentially like running ghost unicast to 30 clients. Also, them being bittorrent clients wont help, because HTTP doesn’t give out different chunks to different clients, all of the clients will have the same pieces – negating the advantages of BitTorrent!

So, if starting a lab of 30 machines: do it 4 machines at a time or something. Get a couple of machines upto a reasonable percentage, so that they can act as seeds to clients that haven’t gotten enough of the pie yet.

3) I can’t build BTDeploy.

Okay – make sure you’ve got the major prereq – the WAIK. Ensure that ImagePE.bat is setup with the correct paths. If the path in the default ImagePE.bat doesn’t have a slash in it, don’t put a slash in yours! If you don’t plan to use Ghost, you don’t need to put Ghost in there. Same with Explorer++, or the DETA builder. You should be fine out of the box, providing you have the WAIK. The other extras are there “just in case”.

4) Some buttons don’t work

As you would probably understand – I don’t code BTDeploy in work time. It’s a project that I’ve started in my own time, so there are obviously sections that are less polished

than others (though, the main bits work). So, if you have a dire need for a specific function – the code is open sourced so you'll be able to do it yourself (and perhaps contribute it to the main source tree). The application is coded in AutoIT, which isn't hard to learn for anyone that's had basic programming experience – and all the tools needed to code the app are actually used in this documentation (Scite4AutoIT3, WAIK, etc). Though, if you sent an email to me, detailing which functionality you needed – theres a fair chance, providing it's not too hard, or unrealistic, that I'd implement it in a reasonable timeframe.