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.

Brent Pickup
2/12/2011

Table of Contents

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.

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.
- 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, you have to ensure that the partitions are setup in the correct order in the WIM file, or, in the "Settings.ini" file in "Program Files\DETA". All indices in WIM files start at '1' [not 0, as is common in most programming languages]. The default layout is set as the following [for a Desktop WIM Image]:

- 1: System [The actual Windows 7 Install]
- 2: Reserved [The 100MB Win7 'Reserved' partition.]
- 3: Data [The Extended 'Data' partition (D: Drive)]

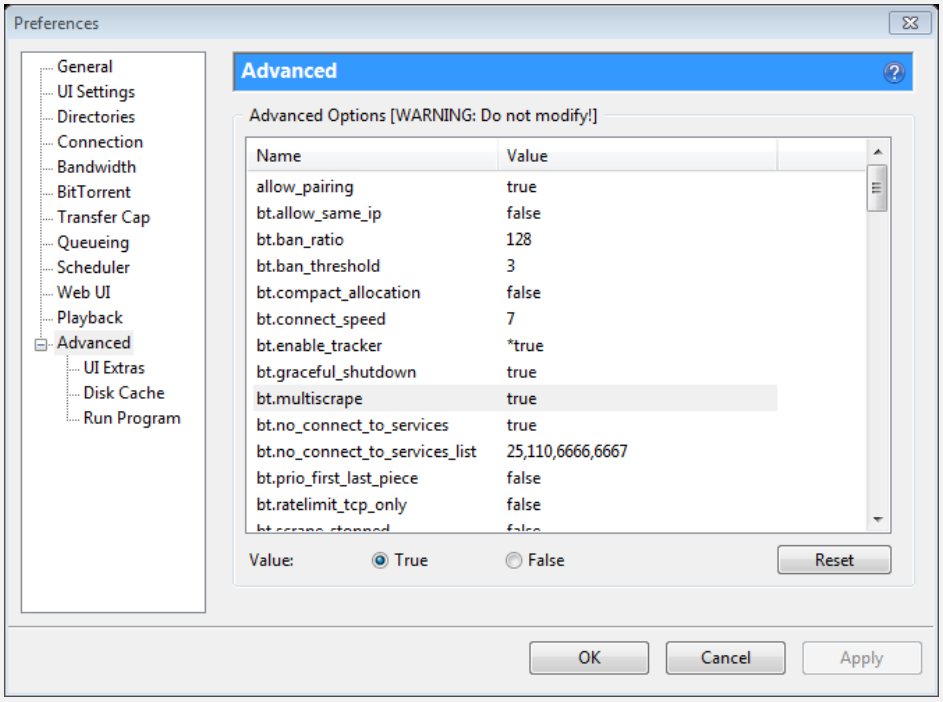
If they are not in that precise order [by default], the software will, for example, apply D: to C:, and vice versa.

Generating your Torrent Files

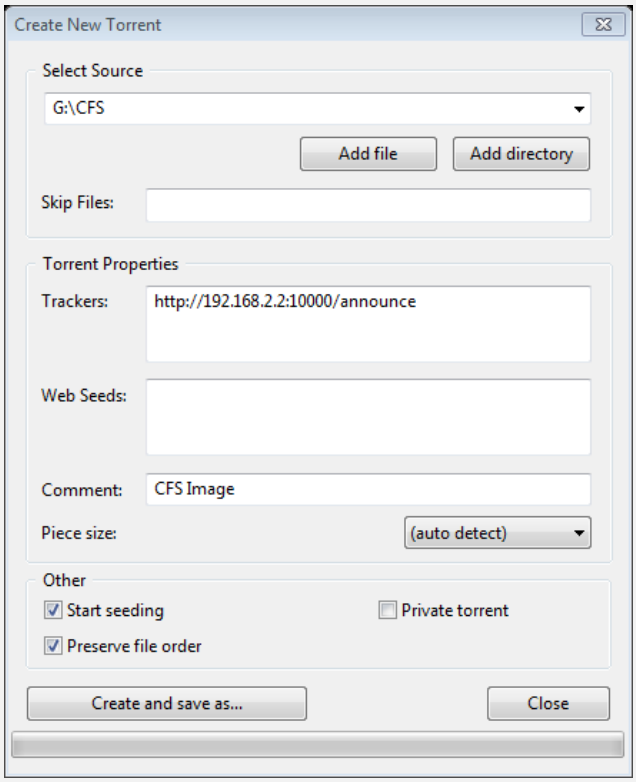
First thing we need to do is generate our torrent files for the install, so BTDeploy has something to actually use. In uTorrent, this is a fairly straightforward setup, and I'm going to assume we have a folder layout something similar to the following for the next step:

- 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 [don't quote me on this – I'm unsure if aria2c – the downloader BTDeploy uses, supports web seeds].

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. 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.
- Grubinst [if you are using my CFS build] in the Program Files folder.

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.

But, back to the "Deployment Path", that you configured before. A sample layout is available at http://drop.edgiest.net/brent/. It has the sample partitioning scripts in there that you'll need to put on your server in the same folders/locations.

Settings.ini

Brief description of each Settings.ini variable:

[Main]
GuiMainLabel=AyrSHS WinPE Environment *Text in main header.*
DeploymentHost=http://drop.edgiest.net/brent/ *Deployment Host. See above.*
[ImageX] *All of these settings pertain to the index in the respective WIM for the specified partitions.*
CFSHomeIndex=3
CFSMOEIndex=1
CFSRecoveryIndex=2
CFSDataIndex=4
DesktopMOEIndex=1
DesktopRecoveryIndex=2
DesktopDataIndex=3
CFTMOEIndex=1
CFTRecoveryIndex=2
CFTDataIndex=3