XashXT Cookbook

Welcome to the XashXT Cookbook! This is a compilation of tutorials from various sources such old archives and other tutorials designed specifically for Xash3D engine and the XashXT modding toolkit! This guide also includes some articles that are a must read for people who want to pursue modding for XashXT.

This guide is created by CRxTRDude as a little archive of knowledge and a makeshift 'documentation' of progress for his mod Nikki Shore and now is sharing them with the community and even to people who aspire to make games with the Xash3D engine.

For further discussion and additional suggestions about the cookbook, see this CS-Mapping thread.

Features of this cookbook

- Mod-making tutorials from basic to advanced designed not only for beginners, but also to pros who are switching to XashXT.
- Recipes in coding and other aspects of modding such as 2D art.
- References and articles for people that can be very helpful.
- Easy to follow navigation and footnotes for additional knowledge.

Introducing XashXT

What is XashXT you might say. From the XashXT help file:

> «XashXT is an extension mod that brings lots of new functionality for mod authors working with Xash3D. Xash3D is an independent game engine that is fully compatible with GoldSource (Half-Life) and has a number of advantages over it.»

Well as it says, XashXT is just a mod for Xash3D, a Russian open-source game engine compatible with GoldSrc, the Quake-derived engine that powered Valve Software's 1998 Game of The Year, Half-Life and subsequent expansions and mods such as Day of Defeat, Team Fortress Classic and Counter-Strike before the release of their more beefier and better Source engine. The engine improves on GoldSrc in many ways, from having extending the engine's limits, to adding new capabilities to the engine, making it a robust and if not better engine than GoldSrc, which is now out-shined by it's superseding engine. The engine is highly compatible with Half-Life, it's single-player expansions and the various single-player mods. Emphasizing single-player, because as of this date, the multi-player functionality is still on the process and will be fixed soon. So far, almost all the functions of Half-Life are compatible to the point that it's not that distinguishable anymore from it's original GoldSrc engine. For more information on the features, check the ModDB feature list here.

The XashXT mod (originally XashMod) is a complete Software Development Kit for Xash3D. Complete, because the mod can also run without Half-Life resources, hence you can make even a full game with just XashXT! This guide or 'cookbook' is just a little compendium on how to add more functionality and stuff to XashXT compiled from various sources. This includes old Half-Life programming sites, TWHL (a very good resource site) and the HLFX.ru forums. This guide just simplifies the process of finding such sources and makes them compatible for XashXT use.

XashXT Feature List

From the original help file:

- **Full compatibility with the original Half-Life entity set.** If you are familiar with Half-Life mapping, then you will feel right at home with XashXT
- Support for forming complex hierarchies of entities with the **Parent System**
- **Mirrors** that can skip reflection of any specified object or, conversely, show ordinarily invisible objects
- **Surveillance monitors** (color and monochrome)
- **Portals** that can take you from one point to another (as in Portal, but more basic)
- **Weather effects** (rain and snow)
- A powerful **particle system** with a standalone visual editor
- **A system of prefixes for making complex scripted scenes**
- An extended **decal projection system** that works on both level geometry and characters (players and NPCs)
- **Realistic lasers** that can be reflected off mirror surfaces (used in gameplay)
- **A large set of logical entities** for implementing any conditions in scripted scenes or complex arrangements
- **A camera with flexible settings** for cutscenes, letting you switch to any character's point of view (including inanimate ones)
- **Configuration options** for setting the attitude of any NPC towards others and towards the player
- **Dynamic colored lighting** that can project and play an animated texture as a light source
- **Playback of AVI files on level polygons and brush objects**
- **A 3D skybox** for simulating expansive areas
- **Bloom effects**
- **Supports for HD textures** (up to 4096×4096)
- **Background maps in the game menu** (as in Source)
- **Support for large game levels** (32768x32768x32768 units)
- **Shadow maps** for dynamic shadows
- **More believable lighting**
- **Realistic rigid body physics** powered by a third-party engine (PhysX)
- **Realistic water**
- **Dynamic day and night cycles**

What you need:

This is a question that you ask yourself first. Well, you of course need Xash3D and XashXT, right? That

is if you want to play mods or just Half-Life but if you want to mod XashXT, you'll need a little extra programs that will help you. Here are the list of things you need to make your mod and play them as well:

- **Xash3D Engine binaries** - You can get them in the ModDB site or the HLFX.ru forums, whichever way, always download the more recent engine binaries. As of writing, the latest one is v0.96, build 2636 [1]. Even if the engine didn't get past version 1, it is still a stable engine to use that you can finish Half-Life with it.
- **XashXT Binaries and resources** - Also available in ModDB or HLFX.ru forums. This is the mod where we'll be working on. As of writing, the latest version is XashXT v0.65. This also contains a Development Kit (xash\devkit) with all the things you need to compile maps as well as the source code that you need to edit.
- **Half-Life Resources (\valve)** - YOU NEED TO BUY THE GAME TO GET IT. Otherwise try finding them in the Internet, good luck with that. Required by XashXT to run. Even if XashXT can be run without Half-Life resources, you need them to run the samples. You can remove the Half-Life resources once your mod is self-sufficient enough to not needing it.[1]
- **Valve Hammer Editor** - There are many links for Hammer in the internet. The editor of choice for Half-Life mapping is also the map of choice for XashXT modding as well. An alternative is Jackhammer, created by XaeroX for Chain Studios, which has the same interface as Hammer, only improved and fixed. Note though that XashXT is not fully compatible with Jackhammer and such is used at your own risk, otherwise, use the Hammer Editor instead.
- **Visual C++ 6.0** - Although I cannot give you a link for VC++ 6 for obvious reasons, you can search the Internet for it. Our main editor for our XashXT source code. If it doesn't work, maybe because you're using Windows 7/8 and want to code in VC++ 6, you can use Code::Blocks, which is an open-source C++ IDE, and incorporate VC++ 6 there. See the article in the Code::Blocks wiki for details on how.
- **Paint program** - Photoshop, GIMP or even Paint.NET, either way, you need one or the other to edit the various textures and art related stuff for Half-Life.
- **Text editor**- Whether if it's the built-in Notepad or any other programmer text editor, as long as you can use it to edit some of the files you might encounter while making your mod. (But please, not word processors. Yes, WordPad is a word processor. Live with it.)
- **Modeling program** - The most common ones used are 3D Studio Max and Milkshape 3D. For more information, see the basic modelling guide for additional programs needed.

Here are specialized programs for Half-Life modding:

- **Wally** - A texture maker program originally for Quake, but is Half-Life compatible. Use this to make textures or import textures made from PS or GIMP.
- **GCFScape** - Primarily for GCFs, which is Valve's proprietary archive for their games, this is also compatible for WADs and PAKs which you will encounter almost everyday of your modding career.
- **PakScape** - One of the only PAK editors out there that supports long filenames, perfect for Half-Life wads.
- **Sprite Explorer** - Used to extract or make sprites for Half-Life.
- Other programs of importance can be found in various places in the internet (if you can find them)

Other programs I recommend

- **Moppi Demopaja** - Originally used for the demoscene, it has a good AVI exporter for making your intro movies and intros, best of all, it's free! (but unstable and requires you to save often if you don't want to waste your work)
- **Audacity** - Open-Source sound editor. Has a slight learning curve, but it's great once you get to know it.

There are more programs that I want to mention, but I couldn't list them all here. I'll try to update this if I can.

Setting up XashXT for modding

Now that you've downloaded a massive heap of ZIPs and EXEs, it's time to put them to good use.

You need to setup a folder where you place your Xash3D engine. I placed mine in my other drive so that C:/ will not be used for anything other than Windows and important stuff that cannot be placed in C:/.

Now that we placed that, follow the instructions from this link. This will get you up to speed in properly installing Xash3D with the Half-Life resources. I assume now that you've finished this step in the process. Now, install XashXT by extracting the zip to the Xash3D installation folder. Your folder should look like this.

Take note of where the Xash.exe and the /xash folder is located. This is to signify that you've installed it correctly. Now that you've done that, there is another step to do before you start modding …

Making your Development Kit

For me, organization is key. Therefore, we need to organize the current development kit and put the stuff we need there.

Step 1. Transferring /devkit to another location

We need this step because it's confusing to place /devkit to the same folder as the XashXT binaries, so we need to MOVE the /devkit folder, particularly just outside the Xash3D installation folder. Here's what my Applications folder (where I installed Xash3D) looked like.
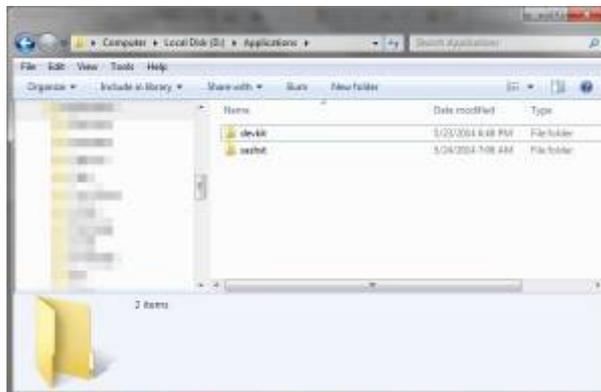
You see that the /devkit folder is there alongside Xash3D. We can now start adding more tools there.

Step 2. Transferring the FGD file to XashXT

This is important especially if we use Hammer. FGD is the file used by the Hammer Editor to associate the entities of the mod to the maps. This step is done in particular for Hammer 3.5 and Jackhammer where you can see the models that it holds and allows you to visualize your map making better.

## Step 3. Setting up the Half-Life tools (Wally, PakScape and others)

Wally and PakScape can be extracted alongside the /devkit folder, preferably in the /tools folder. The Sprite Explorer and GCFScape are installers, so they need to be installed in the computer (although GCFScape has an archive version, it's better to use the installer to make your files associated with the program)

Installing Visual C++ 6.0

If you want to program the mod, you will need Visual C++. Don't worry if it's intimidating to you. The tutorials will make it easier for you to understand coding and programming as you read on through it (after all, I'm actually a nurse by profession. This one is just a hobby. ;)). As usual of installing anything, Installing C++ is no different. You follow the steps, if you're required to type something, you type it (you know what I'm talking about here) and as you progress through installing, you'll be prompted, yadda, yadda. But there are things that you need to know before you install VC++ 6:

• VISUAL C++ 6.0 IS A WINDOWS 98 PROGRAM. This is designed for the old Windows systems and

such, you'll need to run the setup (Setup.exe) as Administrator with Windows 98 compatibility.
- The computer on Windows 7 will always prompt that this will not work in Windows 7. It does actually, but you might need to install Code::Blocks with a few additional steps. See the Code::Blocks wiki for details. The compiler programs WILL WORK on Windows 7, but we cannot assure that the IDE [2] will work.
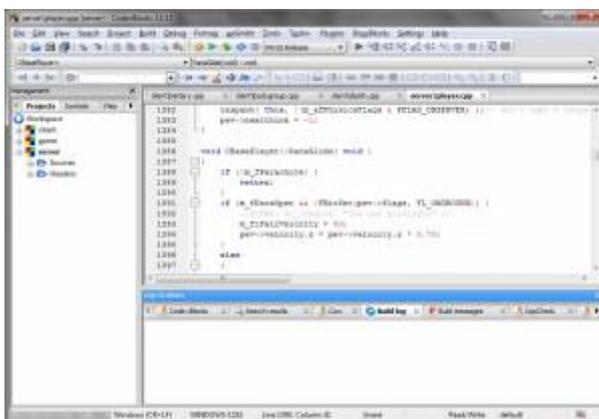
Now the fun starts

From there, you can play Half-Life or other mods. But why stop there? You can continue and start making the mod already now that everything is set-up! You can now continue the tutorials in sequence.

Xash3D Cookbook			
Mod-Making Tutorials	Beginner	Intermediate	Advanced
Recipes	Coding recipes	2D art recipes	
Programs	Mapping	Textures	Resources
References	Half-Life Model Reference	XashXT Entity List	
Articles	Archived	User contributed	

Disclaimer

This is created for the modding community as fair use. Some of the tutorials are acquired from old sources as well as from TWHL. The articles and recipes provided are owned by their respective owners.

Half-Life and GoldSrc are (C) Valve Corporation.
Xash3D is (C) Uncle Mike.
XashXT is (C) The XashXT Team.

[1] NOTE: You will need to update Half-Life to 1.1.1.0 so that it will be highly compatible with your mods
[2] IDE = Integrated Development Environment. A fancy name for a programmer's Notepad. It's just that it's INTEGRATED with the compiler programs and debugging in one place, so you don't have to

search for them.

From:
<http://xash3d.ru/> - **Xash3D**

Permanent link:
http://xash3d.ru/doku.php?id=xashcookbook:en:start&rev=1401952087

Last update: **2014/06/05 08:08**