

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 to the community for 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, the open-source game engine compatible with [W GoldSrc](#) (or GS as we can call it from now on), 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 GS 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 GS, which is now out-shined by it's superseding engine, Source. 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 GS engine. For more information on the features of Xash3D, check out the 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! [XashXT](#) gives the robust features that you seen in Half-Life with better features and capabilities.

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:

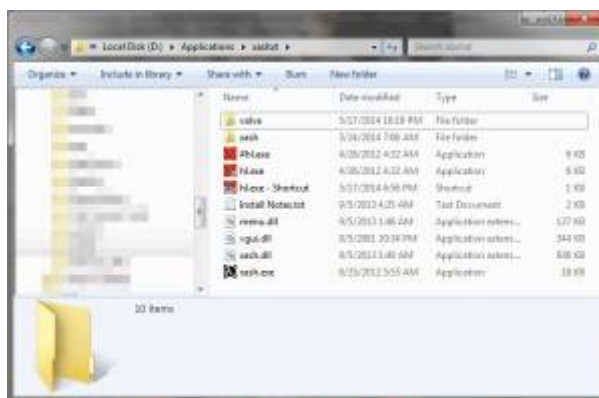
For more information, see [Various Links](#).

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.

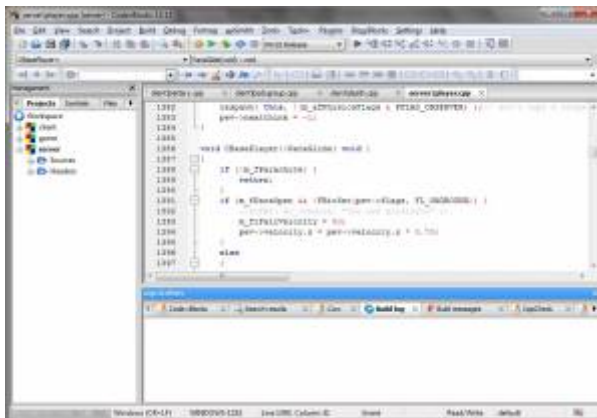
Step 3. Setting up the Half-Life tools (Wally, PakScape and others)

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 ¹⁾ 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. See the bottom bar for details

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	Various Links
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⁾ 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=1402210329>



Last update: **2014/06/08 07:52**