CSave: Installation, Configuration, and Operation

HS Software

1. Installation (Prebuilt Builds) - Windows

This is the recommended way to get CSave up and running. You need to do two things: First, download everything in the build_win64 folder and put all of those files into a folder somewhere. Then, create a folder named "LookupTables" in the folder where you put the CSave executables. Download everything from the LookupTables folder on GitHub and copy them into your LookupTables folder. It is MUCH easier just to use GitHub's "download zip" function to get CSave. That way, you can drag-and-drop the files to anywhere you want. Please note that CSave *will not function correctly* if you do not download the LookupTables properly. Next, open up a command prompt, and use the "cd" command to navigate to where your CSave folder is. It is easiest to open the CSave folder on Windows Explorer, click up top where the path is displayed, and copy and paste the path after the "cd" command on your command prompt.

2. Installation (Compiling) - Unix, Linux, BSD, and Windows (Cygwin)

The first thing you need to do is set up your build environment - install protobuf, gcc, pkg-config, make, and git. After you've installed everything, retrieve the source code for protobuf-c by executing "git clone https://github.com/protobuf-c/protobuf-c.git". Next, enter the source code directory (cd protobuf-c) and execute "./autogen.sh", "./configure", "make -j4", and "make install". If the shell script isn't executable, do a "chmod +x autogen.sh" before attempting to compile. Once you have successfully installed protobuf-c, return to your home directory - execute "cd" - and download the CSave source code with "git clone https://github.com/HackerSmacker/CSave.git". Enter the CSave source code directory with "cd CSave" and compile it with "make". Continue to the Operations section.

3. Installation (Compiling) - Windows (Native)

You need to compile libprotobuf-c with MINGW64 and use the headers/libraries/executables from that and put them into the Lib, Include, and Bin folders, respectively, of your VSC installation.

4. Installation (Compiling) - OS/2 (Watcom C/C++ 11)

Download the source code and transfer it to your OS/2 machine. Please note that you will have to cross-compile libprotobuf-c with GCC targeting OS/2. Transfer libprotobuf-c.lib and the header files over to OS/2. You may need to compile the C protobuf files with GCC, the rest will compile cleanly with Watcom.

5. Installation (Compiling) - OS/2 (EMX/ArcaOS Packages)

Follow the Unix installation instructions written above. The same process should apply. You might need to edit the Makefile to get it to properly compile.

6. Installation (Compiling) - OpenVMS 8.4 (DEC C/C++)

CSave does not cleanly compile with the DEC C compiler. Use a cross compiler or use GNV. Make sure your GCC version in GNV is 8.0.0 or newer, as the protobuf-c outputs will not compile.

7. Operation

7.1. Quick start

Start by running "SaveToProto XXX.sav" and then "SaveUnpack XXX.sav.proto" to get information about a save file. You can then use "SaveGenerate XXX.sav.proto" and then "ProtoToSave XXX.proto" to get a usable file. If you have CSaveGUI installed, please just use that.

7.2. Converting saves

The included utilities "SaveToProto" and "ProtoToSave" to convert to and from "binary format" (i.e. a normal save file) and "protobuf format" (the data actually in the save file) with ease. These programs take one parameter each - the name of the save file.
To convert a save to protobuf format: "SaveToProto original.sav"
To edit the file: "SaveGenerate original.sav.proto edited.proto"
To convert from protobuf format to binary format: "ProtoToSave edited.proto original.sav"

Likewise, you must also convert your profile saves. To acomplish this, use these programs:
To convert a profile to protobuf format: "ProfileToProto profile.sav"
To edit the file: "ProfileGenereate profile.sav.proto profedited.sav.proto"
To convert from protobuf format to binary format: "ProtoToProfile profedited.sav.proto profile.sav"

If you want to convert to and from console saves, simply put the "profile code" number as the last command line argument. For SaveToProto, enter something like this: "SaveToProto 2.sav 3" (3 is the platform code for PS4 saves). For ProtoToSave, enter something like this: "ProtoToSave 2.sav.proto 2.sav 3"

7.3. Converting saves across different platforms

The "SaveConvert" program can convert, for instance, a PS4 save to a PC save. It manipulates the header and re-encodes the save data. Use this program with care, as it is basically a shortcut to running SaveToProto and ProtoToSave, except this program manipulates the headers to make the game not clobber over your save.

7.4. Editing saves

7.5. Comprehensive List of ProfileGenerate commands

7.5.1. set goldenkeys

This command sets how many Golden Keys you have. It will ask for how many you want.

7.5.2. unlock all

Unlocks *everything.* Please note that this could cause some problems. Also, **ABSOLUTELY UNDER NO CIRCUMSTANCES USE THIS COMMAND UNLESS YOU HAVE MET THE LICENSE**

REQUIREMENTS. IT IS ILLEGAL TO USE THIS COMMAND TO OBTAIN CONTENT WHICH YOU DID NOT PAY FOR. I ASSUME NO LIABILITY IF YOU USE THIS.

7.6. Comprehensive List of SaveGenerate commands

7.6.1. quit, exit

Exit from SaveGenerate. The output file will be generated and saved. Please now run ProtoToSave to pack the save file into a BL3 binary format save.

7.6.2. set name

Set the player's preferred name. It will prompt for a string.

7.6.3. set class

Sets the player's class. Takes an integer. 0 is Amara, 1 is FL4K, 2 is Moze, and 3 is Zane.

7.6.4. set sdu

Sets SDU values. Iterates through each SDU and prompts for a new level. Press Enter to use the previous value, or specify a blank line if using a script file. If you are using a script file and you have too many blank lines, the editor will ignore them. If you don't have enough blank lines, the editor will set that SDU to zero.

7.6.5. set mayhemlevel

Set the Mayhem Mode level for any playthrough you want. It will first prompt for the Mayhem level you want, then it will prompt for what playthrough you want to update. Playthrough 0 is NVMH, and 1 is TVHM.

7.6.6. set expoints

Set the total amount of experience points. This does not mean set the level - setting the EXP level affects the level because you need a certain amount of EXP to clear that level. Prompts for an integer value.

7.6.7. set level

Sets the player level by getting the level, and setting the EXP to the minimum required to clear that level. Prompts for an integer. The max accepted value is 80, although this will not be accepted by the game, and will instead drop you down to the current level cap.

7.6.8. set quest

This command will prompt the user for three things: first, the quest path. Use "SaveUnpack name.proto | grep CSAV001MSN" to find the quest. Copy and paste the class path for the mission into the editor. When prompted for the playthrough, enter 0 for NVHM or 1 for TVHM. Next, enter the quest state like this: 0 is Not Started, 1 is Active, 2 is Completed, 3 is Failed, and 4 is Unknown. Do not enter 4, your game will most likely crash.

7.6.9. set guardianrank

This feature is currently not implemented. Check back later for an update.

7.6.10. set money

Set how much money you have. Takes an integer.

7.6.11. set eridium

Sets how much Eridium you have. Takes an integer.

7.6.12. unlock skilltree

Enable the selection of all skills on the tree. Does not coorespond to how many skill points you have.

7.6.13. set skillpoints

Sets how many skill points you have. Takes an integer - there does not appear to be a cap on this value.

7.6.14. set challenge

Modifies a challenge. "Challenges" includes crew challenges, and those challenges that pop up on the left side of your screen every now and then (especially during a new playthrough). Challenges are shared between playthroughs, so it will not prompt if you want to search NVHM or TVHM. It will first prompt you for what challenge you want. Enter the class path of the challenge (remember to use SaveUnpack to find them). Then, enter a completion state (1 for completed and 0 for uncompleted).

7.7. Comprehensive list of CSave message prefixes

CSAV001GEN - General information
CSAV001CLS - Player class information
CSAV001SKL - Skill points, XP, skills, and tree information
CSAV001SDU - SDU information
CSAV001VEH - Vehicle parts, loadouts, and configurations
CSAV001MSN - Missions/quests
CSAV001AMO - Ammo and grenades
CSAV001GRD - Guardian rank, level, perks, and rewards
CSAV001ROM - Crew quarters/bedroom information (including guns on the rack)
CSAV001ECH - ECHO logs
CSAV001FTM - Fast Travel machines: blacklisted, active, and reachable
CSAV001INV - Inventory: backpack and equipped
CSAV001CUS - Customizations: color, skin, emotes
CSAV001CHL - Challenge information
CSAV001ICL - Money (ICL means Inventory Category List)
CSAV001MHM - Mayhem Mode information
CSAVOO1ILT - Item Lookup Test program
CSAV001FIL - CSave file processing messages
CSAV001ABD - Abnormal End (crash)
CSAV001RWS - Read Write Save operations: loading and saving files
CSAV001CNV - Conversion functions

7.7.1. Platform Codes

These numbers are supposed to be entered as the last command line argument on SaveToProto, ProtoToSave, ProfileToProto, or ProtoToProfile.

1 - PC save file
2 - PC profile file
3 - PS4 save file
4 - PS4 profile file