Introduction

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:16:17.921000

Thank You!

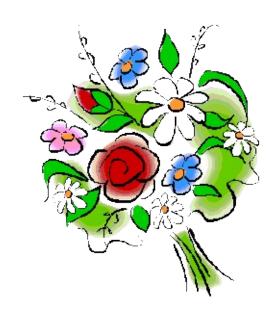
A big "thank you!" goes to everybody, who helped us to create the OpenHoldem software suite and this document:

- AgEmAniaC
- Aiku
- Aintnosunshine
- AlexTheKing88
- AlCapowned
- AleX-YaR
- Amstad
- AndreGoose
- Andrewex
- Angelus
- Ares®
- Azure
- B57inept
- BadHabit
- Barthens
- BBUK
- Bender
- Beutel
- Bibi34
- BillW
- BillyJohn12
- Bitsy
- Blacktie
- Bo858585
- Bob
- Bob4567
- Booster
- BotForSale
- Bozo
- Brdhse1
- BruceLee

- BuckyBall
- Brass Knuckles
- Buchas
- CakeWalk
- Chillit
- ChuckNeverMuck
- Chucksta
- CiderTime
- Cold12zera
- CoolBotMe123
- CrumbleSunday
- Daniel
- DanUtzz
- Darkmund
- Davagin
- Demonthus
- DevBot
- Dimonstrik
- Dipitlow
- Djoca
- Dolar
- DonkeyKong101
- DoyBrungSong
- Dr.Kaktuz
- Eagle_1
- Eden
- EezyMoney123
- Efu
- ElKid
- Epo
- Ess5
- Fasenderos
- Ferfran
- FK1
- FLBotter
- Fn101
- Folder
- FpTrader
- FqFq

- Friday
- FunnyDwag
- Gabber2k
- Ganadai
- Gecko
- Gonhilda
- GoodMood
- Goran123
- Gratiszzzz
- Gtwhat
- Gunfadoso
- GUI
- Guysmo
- Hapil
- Hugo1
- Humanista
- Invader11
- Jabsacyomama
- JConner
- Jesus (our internet friend)
- Jesus (the real one)
- JMonderson
- Joe1111
- JohnNash
- Johnny
- JokeDoke
- JPHol123
- JumpingFences
- KaeKruse
- Kake118
- Karamba1
- Kata
- KiddPoker
- KindBot
- Kkfc
- Knightsaces
- Krishot
- KTrader
- Kyo

- Lavezzi
- LeftOver
- Lucky8
- Macedoniantree
- Mad4IT
- Maniaco
- Marchigno
- Mariano268
- MarlboroKid
- Marquitos
- MasterNe
- Matrix
- Maverick
- MaxInMontreal
- MerzkiyTipon
- Mib92
- Micro111
- Mike
- Mjoe2
- Moby
- Morello
- Mossel
- ot_real
- MrEquity
- MrMosti
- MrRuder
- MustBeatIt
- N4rk0
- Nahman
- Narfal
- NewBot
- Ngels
- Nick Coldhand
- Nick Perry
- Nik0
- NoDBerrHyPn0
- Nomade
- Noobwithquestions
- NPNG



- OpaDorens
- OpenGeek
- OpenHoldem
- OpenHoldemAli
- Oscar
- Papa05
- PastorPoker
- Pawko
- Ppbcon
- Pete
- Pingvvino
- Pippo21
- PkrBot
- Plutus
- Pobar
- Pokerage
- Poker_Dog / TheWhopper
- PokerKid
- pokerweeny
- Postenjak
- Projecthpdv2
- PoundSand
- PyBotter666
- QuiqSilver
- Qwerty
- R2D2
- Radi8tor
- Rambam
- Relhby
- RISA
- Rominus
- Rumburak
- Ryuk33
- Sagan
- Salem
- SamFisher
- Satanikut
- Shank_17

- SingleMalt
- Skyzzo
- Sloboz
- SmartBet
- Smuroff
- SNGChamp
- Spektre
- Spew
- Spud
- StangelNator
- Stars83
- SteveG1963
- Stout-man
- Strikles
- Sturmzealot
- SuperBlak
- SuperTramp
- SweetPork
- Tabo_ger
- Tammelinn
- Termitapalahermita
- TheHighFish
- ThemAndMe
- Thisv8guy
- Tinkerer
- Tobag
- Tobiabus
- Todamach
- Topic7
- Triple
- TripleSeven
- Trisem
- Trish
- Trom
- TT_83
- Ukawtf
- Up_Urself
- USBFreak
- VaDe

- ValueBotty
- Vasyatko
- Vlad2048
- Want2Learn
- WantPlayPoker
- WaveLen
- WetOphelia
- Winlo
- Winngy
- Xtions
- Yep
- ZeeZooLaa
- ZeuZDK
- Zippy
- Zolp
- ZomBee
- ZooBoy

and to everybody we have forgotten. Without your help OpenHoldem and this manual would be incomplete.

Please contact us at www.maxinmontreal.com/forums, if you miss your name on the list.

What is OpenHoldem

OpenHoldem (and its supporting applications: OpenScrape, ManualMode and OHReplay) is an open source screen scraping framework and programmable logic engine for the online Texas Hold'em style poker game. This framework provides the capabilities to allow you to build your own Texas Hold'em robot (bot). There are two major parts to any poker bot: the game state engine and the action engine.

There are a number of approaches for getting the game state from a poker client, including directly reading the memory of the poker software, injecting code into the client's address space, parsing the client's chat box, and interpreting the pixels presented on the screen by the client. OpenHoldem's game state engine uses the last approach — it provides a parameter driven method of interpreting the pixels ("screen scraping"; http://en.wikipedia.org/wiki/Screen_scraping) presented by the poker client to determine the state of the game at any given time. Screen scraping is not a new approach to this kind of problem, poker botting or otherwise. Screen scraping has been a recognized approach for solving a specific set of computer problems almost since computers were invented. Screen scraping has been around for a longer period of time than most of you reading this have been alive!

The action engine in OpenHoldem is called the *Autoplayer*. The Autoplayer will use logic that is provided by you to decide what poker action to take based on the current game state. The Autoplayer will then click the buttons and enter the text on the screen to make the poker client execute this action.

OpenHoldem is a framework. What that means is that you need to provide it with a couple of things to enable the game state engine and the action engine to operate correctly.

The first is the parameters that instruct OpenHoldem how to interpret the pixels presented by the casino that you play at. Each casino presents these pixels in a slightly different manner, and thus these parameters need to be specific for each casino.

Second is the logic to tell the action engine what to do — just like every person plays live/human poker in a different style, this logic will instruct OpenHoldem how to play the style that you want it to play.

OpenHoldem supports every casino in existence today, as far as we know. If you find a casino that does not work with OpenHoldem's game state or action model, please hit the forums and let us know. The developer team is very responsive to

modifying OpenHoldem to work with the continually changing landscape of Texas Hold'em casino client software.

OpenHoldem also supports every sub-flavor of Texas Hold'em poker known to exist today. No-limit, fixed-limit, pot-limit, full ring, heads-up, 6-max, MTT, SNG, double or nothing, turbo, normal speed, whatever. Someone else has already made it work — you can make it work too.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-10T00:24:12.194000

What OpenHoldem is not

OpenHoldem is not a complete poker bot. There is much work you need to do to utilize the framework effectively, however this manual and the community on the forums can help you to do so. If you truly desire a poker bot that includes logic written by someone else, a quick Google search will turn up a number of likely candidates. Be aware, however, that a pre-packaged, complete, click-Setup.exe-and-go, poker bot is not a good way to be successful. Consistent and trusted positive results from these pre-packaged bots are impossible to find. If pre-packaged bots really worked, if we could have spent \$20 to make \$1000's, we certainly would have bought one ourselves, rather than spending the thousands of hours to create this framework!



OpenHoldem does not allow for automated collusion. Poker botting currently is in a gray area of legality and morality, with many strong perspectives on both sides of the argument. The OpenHoldem development team strongly believes that automated collusion is clearly illegal and immoral, and will not provide the facility to help you achieve these ends.

OpenHoldem is not a general-purpose poker botting engine for all styles of poker. It was intentionally suited only to the "Texas Hold'em" flavor of poker favored by the majority of online players today, and at present it supports "four-card-Hold'em" also. But don't ask if it supports Stud or Razz — it does not.

↓WinHoldem heritage

Reproduced from the original post by Matrix (http://www.maxinmontreal.com/forums/viewtopic.php?p=28468):

"For the first couple of years of its existence WH (WinHoldEm) was actively developed and developed an enthusiastic following. Although it was a commercial product Ray Bornert put in an astonishing amount of time answering queries on the WH BBS, and seemed to respond to opinions and requests there pretty quickly. One of the features he introduced were private forums which groups could request and then run their own discussions and developments within them. These must have been problem children for him. because there was a measure of falling-out, squabbling and bad blood within them. The biggest and most active of these was the infamous Lab1.

The first sign of Ray's movement away from the total nurture and support role was the introduction of Winscrape, a utility to enable the creation of profiles (Table maps). In principle WH users were then not completely dependent on him to generate table maps. This was probably motivated by the increasing tendency of pokersites to change their client display slightly at frequent intervals, and maintaining table maps must have been an increasing burden for Ray. After a period he then stopped creating table maps altogether, and threw the users onto their own abilities with Winscrape. A market economy around them was supposed to develop. There were external WH groups by this time. Ray had closed down the labs, saying that he did not want to be in the position of one who could be accused of spying on their work, and forced them to find alternative homes. The only one which did this permanently and successfully was Lab1; ending up on a host run by a mysterious presence called MaxInMontreal. The external market economy for table maps did not work very well, because Ray had put no mechanism in WH to enable maps to be given an equivalent of DRM, and piracy and strife soon occurred. These three threads combined to generate a lot of dissatisfaction amongst the dedicated WH users. The focus of this was in Lab1. The practical result was that a certain element produced a WH functional clone to seed an open source project called OpenHoldem. An open source equivalent of

Winscrape followed.

This gamble seems to have paid off. There has always been a dichotomy in botters between their interests in collaborating to develop techniques, and their interest in protecting the final commercial results - the bots. Lab1 helped enormously by making their table maps non-commercial. This established the precedent that the infrastructure for a bot was a community responsibility and development. People seem to have accepted this as a good working model. As to the hatred — the major OH protagonists acknowledge their debt to Ray for WH, and the only comment that Ray has made about OH is that you cannot compete with open source."

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-10T00:23:37.803000

A (few) word(s) of caution

Effort

This is a lengthy manual, and for good reason. This is a complicated hobby with a steep learning curve. If you are reading this and coming into this hobby for the first time, and you have the expectation of running a profitable poker bot tomorrow, then we kindly suggest you close this manual and look elsewhere. This hobby is very rewarding, intellectually, academically and monetarily, but that reward does not come without effort. Maybe one of the \$20 pre-packaged bots will meet your needs, if truly all you want to do is to MakeMoneyFast. You will find a list of interesting software for example at http://www.pokerstars.com/poker/room/prohibited/.

Scams

There are people who will try to sell you partial or complete OpenHoldem-based poker bot solutions. The same warning applies to these offers as to the \$20 click-Setup.exe-and-go poker bots. Ask yourself how much you would sell a complete poker bot for if it was profitable and made you money every day, even while you were sleeping? A simple calculation should clarify this point:

- Say you have a pokerbot that makes \$2 per table per hour.
- Most sites allow you to run 4 tables simultaneously without issue. This equates to \$8 per hour.
- Further assume you run your bot 6 hours per day to better model human play. That gives you \$48 per casino per day.
- If you then run your bot 5 days per week, you stand to make \$12,480 per year.

Multiply that by the number of casinos you play at, and you have a reasonable target selling price. Would you be willing to pay that amount of money for an unproven, "complete", "winning" poker bot? How much do you think a \$100 or \$500 "complete" poker bot is really worth?

There is a long history of people being scammed out of hard earned money for "winning" poker bot solutions. Sometimes lots of money. Don't be stupid...the age-old maxim applies here: if it sounds too good to be true, it probably is.

Gray area

The hobby of poker botting is arguably a legal and moral gray area. While other online real money activities actively allow and even encourage the use of bots (stock trading, currency speculation, sports betting, etc), online poker has taken a somewhat unusual stance on this topic. Some casinos will attempt to detect your bots with various means, including spyware-like actions of scanning your hard drives and taking screen shots, statistical analyses of mouse clicks, post-analyzing hand actions for consistent behaviors and so on.

What this means, however, is that one must take caution when poker botting to ensure that you protect yourself, your money balances, and your private information on your computer. All of these anti-bot detection attempts can be defeated, and this will be discussed further in the stealth section.

On the legality question, the legality of poker botting has not been court-tested to date. Even if it was, some of these casinos believe they operate above the law and can do whatever they want to your account at any time.

Elaboration on the fine points of these arguments is beyond the scope of this document, but a vigorous discussion can be found on forums online, including PokerAI (http://pokerai.org/pf3/viewforum.php?f=80). The truth is that as long as casinos need to display the game state on your computer screen, there will be a way to use a bot to play the game. Simple fact.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-29T21:54:48.875000

References

OpenHoldem References

- Discussion forums: http://www.maxinmontreal.com/forums
- Binary package download site: https://github.com/OpenHoldem/openholdembot/releases
- License, GPL v3: http://www.gnu.org/copyleft/gpl.html
- Source code: https://github.com/OpenHoldem
- Old source code (OpenHoldem 1.0 ... 7.7.3): http://code.google.com/p/openholdembot/source
- Source instructions: http://www.maxinmontreal.com/forums/viewtopic.php? f=113&t=3542

Other open source projects used in the OpenHoldem project

- Spirit parser library; http://www.boost.org
- HScrollListBox; Nebula Technologies, Inc.; http://www.nebutech.com
- Dynamic window resizing class; Paul DiLascia; MSDN Magazine July 2001
- PokerEval library; https://gna.org/projects/pokersource/
- Hashing algorithm from Bob Jenkins; http://burtleburtle.net/bob/hash/doobs.html
- Visual Leak Detector; http://dmoulding.googlepages.com/vld
- Normalized random distribution Java class; Indiana; http://pokerai.org/pj2/index.php
- Scintilla source code editing component; http://www.scintilla.org
- Scintilla MFC C++ wrapper class; Horst Brückner; http://www.codeproject.com/editctrl/scintillawnd.asp?select=1475770
- Progress Indicator Dialog; P. GopalaKrishna; http://www.codeproject.com/win32/UPDialog.asp
- Netscape-style preferences dialog; Chris Losinger; http://www.codeproject.com/KB/tabs/saprefs.aspx
- Fast median search algorithm; Nicolas Devillard; Niklas Wirth; http://ndevilla.free.fr/median/median/index.html
- A lightweight RPC library based on XML and HTTP; http://xmlrpc-c.sourceforge.net/

- A string-CRC-algorithm; http://www.kernel.org/doc/Documentation/pcmcia/crc32hash.c
- MD5-checksum for files by Shaheer Abdulrahiman and "Napalm"; http://www.codeproject.com/Articles/32202/MD5-File-Creation-and-Verification & http://www.rohitab.com/discuss/topic/21740-md5-hashing/
- Mersenne random algorithm by Makoto Matsumoto and Takuji Nishimura; http://dmorse.github.io/simpatico/doc/html/mtrand_8h_source.html

Image sources

• http://40.media.tumblr.com/tumblr_lo17fqFYal1qj5qvfo1_500.jpg

Recommended tools

- PokerTH, an open-source poker simulator: https://www.pokerth.net/download
- DDPoker, a freeware poker simulator: http://static.ddpoker.com/download/
- GNU CoreUtils, including head and tail to handle huge logs: http://gnuwin32.sourceforge.net/packages/coreutils.htm
- GNU grep to filter log-files: http://gnuwin32.sourceforge.net/packages/grep.htm
- AutoIT, a freeware scripting language for tools like hoppers: https://www.autoitscript.com/

How OpenHoldem works

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:34:21.593000

Connecting to a table

As soon as you start OpenHoldem it will read all tablemaps in your scraper-folder and parse them for the information that is necessary to connect to a table. The auto-connector will then have a look for new tables that are not yet served by other instances once per heartbeat and connect if it finds a new one. This all happens automagically, however an explanation of what happens behind the scenes is useful if you have problems connecting to your target poker window. The following steps are taken to identify which windows on your screen are valid poker tables.



1. A list of all visible top level windows that have a non-blank caption is collected

- 2. Each window in this list is compared against each tablemap that you have placed in your *scraper* directory. For each step that follows, if a non-match is detected, OpenHoldem will then immediately move on to the next window / tablemap.
- 3. The size of the client portion of the window is compared to the *clientsize* records in the tablemap. The window is considered a match if the tablemap indicates the correct size as given by the record clientsize, or if the size is within the size limits given by *clientsizemin* and *clientsizemax*. Example from a real-world-tablemap:

```
//
// sizes
//
z$clientsizemin    400  300
z$clientsizemax    1600  1200
z$targetsize    800  600
```

4. The window's title text is compared to the keyword text filters provided by *titletext* records in the tablemap. The window is considered a match if the free-form text in any of the titletext records is found in the window's title. Additionally, if any negative *!titletext* records are present in the Table Map, then a window is considered a non-match if the free-form text in any of the negative !titletext records is found in the window's title. Example:

s\$titletext PartyPoker

- 5. Optionally OpenHoldem will also compare tablepoints. These tablepoints are useful to distinguish very similar tables that require different tablemaps, for example 10-chair and 6-chair-tables. They can be distinguished by placing a tablepoint at one of the chairs that does not exist for the other kind of table.
- 6. If OpenHoldem finds a matching pair of tablemap and table it will look in the shared memory data-structures if the table gets already served by another bot. If this is not the case OpenHoldem will connect to that table. The connected table will then be moved to a free slot if the table-positioner is enabled, resized if a targetsize is specified and the autoplayer will start playing.

The Heartbeat-Cycle

Once OpenHoldem has been connected to a table, it will enter into a continuous loop that performs a variety of actions. The cycle of these actions is described here in order of their execution.

Action Explanation

Ask the *screen scraper* engine to scrape and interpret the

current pixels on the poker window based on the

Scrape window parameters in the loaded tablemap. The scraper will look

for example if a chair is occupied, if the player is sittong out, what the players betsize is and so on. It then will hand

over this information to the game-state-engine.

OpenHoldems built-in *scraper preprocessor* will then

preprocess some textual input from the scraper; especially title-string, bets and balances. For example it will remove

spaces inside numbers, replace commas inside numbers by dots, replace outlandish currency-signs by dollars or

remove other superflous texts, such as "pot: (150 + 300)".

Create replay If a replay frame is requested in preferences, OpenHoldem

will store an image file and a HTML-file in the replaydirectory for a future settion-review with OHReplay

Calculate symbols Calculate the built-in OpenHoldem *symbols* that can be

accessed by the user-defined poker-logic.

If the *validator* is enabled in preferences, then it will check

Validate game state the current game state for inconsistencies that are usually

caused by mis-scrapes or bad tablemaps.

If it is our turn and the scraper-engine saw enough *stable frames* then OpenHoldem will evaluate the set of formulas

that got provided by the user.

Evaluate the bot-

logic

Scraper

frames

preprocessor

• OH-script: f\$alli..f\$fold

• OpenPPL: f\$preflop..f\$river

This user-defined bot-logic usually resides in a plain-text file with the extension *.ohf (OpenHoldem formula), *.oppl

(OpenPPL) or *.txt (also OpenPPL)

Handle spam

Evaluate any i86 regions, and dismiss any popups that occlude the table.

If the autoplayer is engaged, then OpenHoldem will act: press buttons, enter betsizes, move the slider. The autoplayer will try to execute the most aggressive actions first, then continue with less agressive ones, i.e. in the following order:

- f\$alli
- f\$betpot_2_1 .. f\$betpot_1_4
- f\$betsize
- f\$rais
- f\$call
- f\$check
- f\$fold

Autoplay

The autoplayer works — as the name suggests — automatically, but the necessary actions differ a bit for each casino, therefore the autoplayer can be customized with the tablemap. For example you can tell how an old betsize should be selected before a new one can be entered:

s\$betsizeselectionmethod

Click Drag

Wait

We will deal with that in the chapter about tablemaps. Sleep for some time, then start the next cycle. The heartbeat-sleeping-time can be specified in the preferences and defaults to 750 ms. However the heartbeat delay is somewhat flexible: we scrape faster if it is our turn to act for faster reaction times and we sleep longer when we have no cards (folded) or if we are not seated at all to optimize CPU-usage.

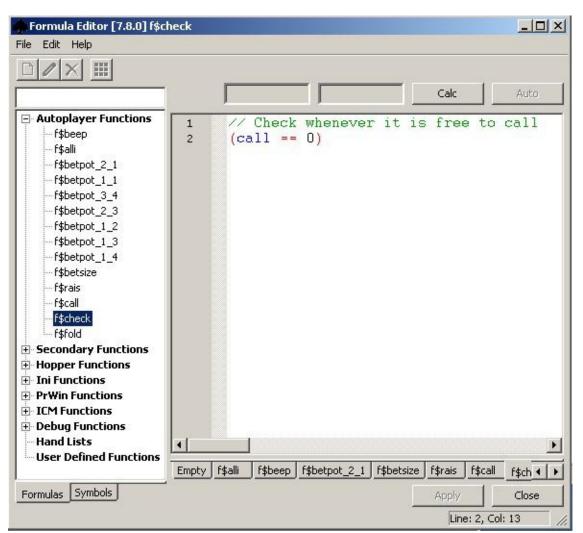
The Bot Logic (OH-script)

OH-Script

Every bot needs some poker-specific logic to decide how to act. OpenHoldem provides a basic scripting-language for that, which was inherited from WinHoldem. The logic is action-centered, i.e. everytime it is your turn the following basic questions have to be answered:

- "shall I go allin?" f\$alli (boolean, interpreting 0 as false and everything else as true)
- "shall I click a betpot-button?" f\$betpot_2_1...f\$betpot_1_4 for twice betpot to 1/4 betpot.
- "shall I raise, and how much?" f\$betsize
- "shall I (min-)raise" f\$rais (mainly for Fixed Limit; boolean)
- "shall I call" f\$call (boolean)
- "shall I check" f\$check (boolean)
- "shall I fold" f\$fold (boolean)

Per default f\$check evaluates to true if it is free to call and f\$fold is always true, so if all questions were answered with 0 or false, your hand-will be checkfolded.



OH-script uses C(++)-style mathematical expressions, i.e. the same mathematical operators like the C-programming-language and OpenHoldems built-in symbols. An example:

The DLL-extension

Every bot needs such a formula file, even if it is programmed in C++. There are no standalone C++-bots; the user-DLL is just a programming extensions that make it possible to extend the functionality of OpenHoldem by providing additional symbols (constants and parameter-less functions). The very minimum is a small OH-script or OpenPPL formula that redirects every primary function to a DLL-function.

```
##notes##
// Very small OH-formula, that redirects
// primary functions to C++-functions in the user.DLL.
##f$alli##
dll$alli
##f$betsize##
dll$betsize
##f$rais##
dll$rais
##f$call##
dll$call
```

In practice formulas can become arbitary complex, but those are the basic principles you will need to understand. We will deal with the details of bot-programming in the chapter about OH-script.

Parse-Time and Run-Time Errors

The OpenHoldem scripting language will detect various error conditions for you and warn you of these conditions. Each time a formula is loaded, it is "parsed" for correctness – if there are any syntactical errors upon this parse, then a message box will be displayed that alerts you to the location of this error. Additionally, several classes of run-time errors will be detected, including divide-by-zero and infinite recursion. Note that the OH-Script language does not support recursion in any form, due to the absence of the assignment operator and function parameters to control the recursion depth.



Default Bot

All OH-script-bots are self-contained. Once all questions f\$alli .. f\$fold have been answered the best action is well-defined. Therefore OH-script does not support a default-bot. "All power (and responsibilities) to the user" is the motto of OH-script. Things are slightly different for OpenPPL, the other programming language, however. An OpenPPL-bot might be "incomplete". The end of a code-section might be reached without a matching action being found, then the default-bot will take over. We will talk about OpenPPL and this default-bot in the next chapters briefly; all the details are in an extra handbook, the OpenPPL-manual.

The Bot Logic (OpenPPL)

Since version 7.0 OpenHoldem provides an alternative way to program the botlogic: OpenPPL. PPL means: Poker Programming Language, an english-like programming language once designed by Shanky BonusBots. OpenPPL is completely integrated and can be mixed with OH-script if desired. Once a non-empty f\$preflop function is present OpenHoldem will switch to OpenPPL-mode and evaluate f\$preflop, f\$flop, f\$turn and f\$river instead of the OH-script autoplayer-functions.

```
##f$preflop##
WHEN hand$AA RaiseMax FORCE
WHEN hand$KK AND Raises = 0 AND Calls = 0 RaiseTo 3 FORCE
```

To support an easy start we wrote a separate OpenPPL-manual that is included in every download of OpenHoldem.

Default Bot

An OpenPPL-bot might be "incomplete". The end of a code-section might be reached without a matching action being found, then the default bot will take over. We will talk about this default-bot in one of the next chapters briefly; all the details are in an extra handbook, the OpenPPL-manual.

Backup Actions

In some situations it might be not possible to execute the users preferred action. For example he chose bet 3/4 pot, but the casino does not provide a button for 3/4 pot. Then he usually wants to execute a similar action, in this case betpot ot bet 2/3 pot. We call this a *backup-action*.

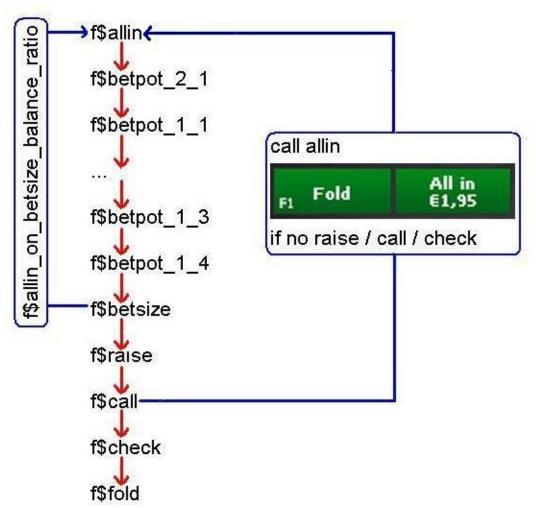
OH-Script and Backup-Actions

Long story short: OH-script does not provide any *built-in* backup-actions. The OH-script-philosophy is: full control to the user. Therefore the end-user has to care about backup-actions on his own and add e.g. a rule that makes f\$betpot_2_3 true if f\$betpot_3_4 was true but could not get executed.

```
##f$betpot_2_3##
// backup for betpot_3_4
f$petpot_3_4
OR ... // rest of logic for 2/3 betpot
```

OpenPPL and Backup-Actions

OpenPPL backup-actions



OpenPPL is different: there is no way to control each possible action separately when you specify what to do on preflop, flop, turn or river. Only one action can be chosen and simplicity was an important design-goal for OpenPPL. Therefore backup-actions are built-in and work automatically. Most of them work the conservative way: if one action fails OpenHoldem will choose the next, slightly less aggressive action: so if betpot fails OpenHoldem will try to bet 3/4 pot, but not twice the pot or even allin.

Aggressive Backup-Actions

Most backup-actions are conservative, but there are two aggressive exceptions:

• *f*\$allin_on_betsize_balance_ratio: playing poker it happens regularly that a

preferred betsize is nearly allin. Then it usually does not make sense to bet that amount and keep only very few chips left. We would be committed anyway the next time we act, and with a larger bet we generate a little bit more fold-equity. The function <code>f\$allin_on_betsize_balance_ratio</code> controls how large a bet has to be in comparison to the stack to turn the betsize into an allin-action. If the value is 0.50 (meaning 50%) any bet that is larger than 50 percent of our stack will cause an allin. This backup-action can be used in OH-script too, because it can be completely controlled by the user.

• *calling allin*: in some cases it might be that an opponent puts us allin and only calling and folding are possible. But our hero looked at his fine set of aces and decided to "bet 2/3 pot for value". OpenHoldem ended up folding. What did happen? Well, some casinos don't display a call button in this case, but only allin and fold. The conservative sequence of backup-actions therefore will finally (and successfully) try to fold. To make all the careless newbies happy we therefore created an aggressive backup-action that turns a call (or any raise) into an allin-action if only the two buttons fold and allin are visible.

Attention! Danger!

Automatic beackup-actions are convenient. But the last example also illustrates some danger. Let's assume that hero does not have a set of aces, but a busted hand. He reasons that the board does not hit villains range either, therefore he goes for a poor bluff and decides for 2/3 betpot. Unfortunately villain already raised allin and hero can only fold or call. Instead of a 2/3 potsized raise he will call allin — with air.

Truly: this is dangerous, but we can't make both users happy at the same time, so we took care about the most common use-case: newbies who hold some value and really want to put money into the pot, but don't consider if villain is already allin and don't consider the options they really have. Please always remember: *There is no backup for your own thoughts and for your carefulness.*

Hopper functions

Once every four seconds OpenHoldem will evaluate and execute some hopperfunctions.

Function **Explanation**

OpenHoldem will click the sitin-button (i.e. the button witl f\$sitin

if f\$sitin evaluates to true.

f\$sitout OpenHoldem will click the sitout-button if f\$sitout evaluat f\$leave

OpenHoldem will click the leave-button if f\$leave evaluate

OpenHoldem will click the [x] in the titlebar of the table if f\$close

evaluates to true.

OpenHoldem will click the rematch-button if f\$rematch ev f\$rematch

OpenHoldem will click the autopost-button if f\$autopost e f\$autopost

true.

OpenHoldem will execute a user-defined rebuy-script if f\$: to a positive target-amount (in dollars) and safety constrain met. For the details please see the chapters about automatic

the preferences dialog for rebuys.

OpenHoldem evaluates this function once per handreset an formula-file that got selected by the loadformula\$XYZ-syr file is expected to be in the bot_logic-directory. Example:

##f\$select_formula_file##

f\$select formula file

f\$rebuy

WHEN (StackSize < 15) RETURN loadformula\$push_f

WHEN issng RETURN loadformula\$my_sng_bot\$ohf FO

Log-Files

Sometimes the bot might take an unexpected decision, more rarely for bots in production, but for sure regularly for bots in development. The reason can be mis-scrapes (Table Map or improper system settings); it can be bad bot-logic; it can even be a bug in OpenHoldem.

Many newbies will automatically assume the latter case. Whenever you encounter a strange behaviour it is a good decision to make use of OpenHoldems logging capabilities. OpenHoldem will generate a log file as it operates, less detailed logs for production, more detailed logs for debugging. The verbosity of the log can be set in the Preferences tabs *Logging* and *Debugging*. The log file can be found at *OpenHoldem\logs\oh_NN.log* where NN is the session-ID.

Basic Info

On every autoplayer action OpenHolldem will create a section with some basic info about the table-state, the decision taken and the action executed.

```
2015-09-10 08:17:35 - Version: 7.8.0
2015-09-10 08:17:35 - Chairs: 9
2015-09-10 08:17:35 - Userchair: 4
2015-09-10 08:17:35 - Holecards: 7d9c
2015-09-10 08:17:35 - Community: ......
2015-09-10 08:17:35 - Handrank: 2
2015-09-10 08:17:35 - Hand: hcard
2015-09-10 08:17:35 - My balance: 2135.00
2015-09-10 08:17:35 - My currentbet: 80.00
2015-09-10 08:17:35 - To call: 0.00
2015-09-10 08:17:35 - Pot: 280.00
2015-09-10 08:18:19 - Big blind: 80.00
2015-09-10 08:18:19 - Big bet (FL): 160.00
2015-09-10 08:18:19 - f$betsize: 0.00
2015-09-10 08:18:19 - Formulas: FCK..
2015-09-10 08:18:19 - Buttons: F.KRA
2015-09-10 08:18:19 - Best action: f$call
2015-09-10 08:18:19 - Action taken: f$check
```

As you can see in this example the bot-logic decided to call, but there was no bet to call and no call button either. So OpenHoldem finally checked (backupaction).

Autoplayer Trace

If you want to go into detail the most useful part of the log is clearly the autoplayer-trace. It helps you to investigateö the entire evaluation that caused a decision. Let's assume we have a f\$alli-function that looks like that:

```
##f$alli##
   f$HaveUnbeatableMonster
OR f$HaveSuperDuperDraw
OR f$PoorHeroIsPotCommitted
```

On evaluation of f\$alli OpenHoldem will look up the three user-defined functions (f\$HaveUnbeatableMonster, ...), evaluate them and combine their return-values to a final result. This leads to a simplified call-tree:

```
fSalli = 1.000 [Line 2/1791]

f$HaveUnbeatableMonster = 0.000 [Line 18/3976]

f$HaveSuperDuperDraw = 0.000 [Line 68/574]

f$PoorHeroIsPotCommitted = 1.000 [Line 2/1837]
```

Each function that is one level indented got called by the next function above that is one level less indented. Things would be easy (or a night-mare) if all code was in one oversized function and all called symbols were black boxes. However one of the advantages of OH-script and OpenPPL is that you can structure your code and create functions on your own. Indeed OpenPPL provides several hundred poker-logical functions that work out of the box. They are all available as plain text, so that you can have a look at their code and OpenHoldem can log their evaluation in detail. Let's assume your user-defined function f\$HaveUnbeatableMonster is an easy one and just considers the winning percentage (the built-in symbol prwin), f\$HaveSuperDuperDraw uses the OpenPPL-symbols HaveStraightDraw and HaveFlushDraw, whereas f\$PoorHeroIsPotCommitted uses Balance and Pot. Then the autoplayer-trace could look like this:

```
f$alli = 0.000 [Line 2/1791]

f$HaveUnbeatableMonster = 0.000 [Line 18/3976]

prwin = 0.731

f$HaveSuperDuperDraw = 0.000 [Line 68/574]

HaveStraightDraw = 0.000 [Line 12/719]

HaveFlushDraw = 0.000 [Line 5/413]

f$PoorHeroIsPotCommitted = 1.000 [Line 2/1837]

Balance = 12.000 [Line 7/857]

Pot = 86.500 [Line 3/3088]
```

So the implementation and the evaluation of each symbol use some other symbols that are one level deeper indented in the autoplayer-trace. In practice the autoplayer-trace will get even more detailed. For example even a simple symbol like Balance might need the user-chair, the size of the big-blind and some more. But the principle is always the same and with a little bit experience the logs become easy to read.

OpenPPL

OpenPPL works very similar to OH-script. Basic info and the autoplayer-trace are the same.

Initializations

Most OpenPPL-symbols get calculated on the fly when you need them. But some symbols require special initialization, exactly once per orbit when it is our turn and the scraper saw enough stable frames. This is the case mainly for counters like "Raises", "Calls" and some more.

```
InitMemorySymbols = 0.000 [Line 18/3976]
  TimeToInitMemorySymbols = 1.000 [Line 8/3921]
   isfinalanswer = 1.000
   GameStateChanged = 1.000 [Line 2/3928]
        dealerchair = 4.000
        me_re_MemOldGameStateDealerChair = 3.000
   False = 0.000 [Line 1/230]
   InitMissingSmallBlind = 0.000 [Line 2/772]
        betround = 1.000
        ConstBetRoundPreflop = 1.000 [Line 1/199]
```

You will see these initializations on top of the autoplayer-trace, i.e. below the

summary, but above f\$preflop.

Main evaluations

After everything got initialized OpenHoldem will evaluate the main OpenPPL-function for the current betround: f\$preflop, f\$flop, f\$turn or f\$river.

```
f$preflop = 11.000 [Line 93/2726]
  StackSize = 26.688 [Line 1/1422]
  StackSize = 26.688 [cached]
  BigBlindSize = 80.000 [Line 1/1373]
    bet1 = 80.000
  BigBlindSize = 80.000 [cached]
  listGp1 = 1.000
  listGp2 = 0.000
  listGp3 = 0.000
  user_EarlyZone = 1.000
  Raises = 1.000 [cached]
  Calls = 0.000 [Line 1/630]
    me_re_MemCalls = 0.000
  Calls = 0.000 [cached]
  BotsLastAction = -1000007.000 [Line 2/573]
    None = -1000007.000 [cached]
```

Return Values

You might wonder, what the numbers mean:

```
f$preflop = 11.000
...
BotsLastAction = -1000007.000 [Line 2/573]
None = -1000007.000
```

The evaluator-engine needs to handle three different type of final actions:

- betsizes (in big-blinds), encoded as positive numbers, so f\$preflop = 11.000 in the example above means RaiseTo 11 (big blinds).
- percentaged potsized bets, encoded as small negative numbers. -0.666 means bet 2/3 pot.
- fixed actions, encoded as large negative numbers.

```
##Beep##
-1000000
##Fold##
```

```
-1000001

##Undefined_BetSize_Probably_Due_To_Misscraped_Pot_Defaulting_Tc
-1000002

##RaiseMin##
Raise
```

Altogether 25 named action-constants — too many to mention them here. But they are in the OpenPPL-library which is a plain-text-file. Feel welcome to have a look — there are many interesting things to discover.

Undefined Actions and Return Values

Whenever OpenHoldem reaches the end of a function it will evaluate a special symbol "empty_expression__false__zero__when_others_fold_force". So it will no longer continue silently with an "undefined" value like in the past. Instead it will create a verbose log-file. This symbol always evaluates to zero, which is also the encoding of "False" and "Fold".

```
f$flop = 0.000 [Line 27/73]
...
  empty_expression__false__zero__when_others_fold_force = 0.000
f$check = 1.000 [cached]
f$fold = 1.000 [cached]
```

So whenever you see this symbol in your log-files you know that your profile does not handle the current situation. This regularly happens for so-called "tweaks", i.e. incomplete profiles, meant to be used on top of another one. Folding a set of kings? No wonder, nothing to be tweaked, playing sets of kings is easy. In such cases you have to complete your profile. Tweaking, completing and combining profiles is explained in the OpenPPL-manual.

Autoplayer Actions

After the OpenPPL-decision gets calculated OpenHoldem translates it to OH-script autoplayer-actions (plural because of backup-actions) and thereafter the autoplayer works the normal way. That's why you can see some OH-script autoplayer-functions at the end of the autoplayer-trace, each one with a precomputed ("[cached]") value. Please be aware that OpenHoldem — contrary to OpenPPL — doesn't use big-blinds for its calculations but dollars. So "RaiseTo 11" in the example above leads to a f\$betsize of \$880.

```
f$betsize = 880.000 [cached]
f$rais = 1.000 [cached]
f$call = 1.000 [cached]
f$check = 1.000 [cached]
f$allin_on_betsize_balance_ratio = 0.000 [Line 5/1864]
f$fold = 1.000 [cached]
```

Errors and Warnings

Besides of the autoplayer-trace OpenHoldem logs all errors and warnings, i.e.:

- all shown error-messages
- all message-boxes that got suppressed by your preferences
- some more errors and warnings where an error-dialog would not be appropriate.

Debug Messages

Debug messages are mainly meant for developers and for experienced users who want to get more info about the way how OpenHoldem works internally. These messages can be turned on with the Debugging-tab of the Preferences dialog. They are very useful for finding and fixing bugs, but they are usually not needed for most end-users. Below an example about blind-detection, blind-locking (for stability) and detecting hand-resets.

```
2015-08-15 16:00:24 - [CHandresetDetector] Setting handnumber to []
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by dealerchair:
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by cards: false
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by handnumber:
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by community ca
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by potsize: fal
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by nopponentspl
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by increasing b
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by changing bli
2015-08-15 16:00:24 - [CHandresetDetector] Handreset by new small bl
2015-08-15 16:00:24 - [CHandresetDetector] Methods firing this heart
2015-08-15 16:00:24 - [CHandresetDetector] Methods firing last 3 hea
2015-08-15 16:00:24 - [CHandresetDetector] Number of methods firing
2015-08-15 16:00:24 - [CHandresetDetector] Handreset found
2015-08-15 16:00:24 - [CHandresetDetector] Setting last dealerchair
2015-08-15 16:00:24 - [CSymbolEngineTableLimits] CalcTableLimits()
2015-08-15 16:00:24 - [CSymbolEngineTableLimits] ResetOnHeartbeat() 2015-08-15 16:00:24 - [CSymbolEngineTableLimits] TableLimitsNeedToBe
2015-08-15 16:00:24 - [CBlindGuesser] Data from scraper (ttlimits, c
```

```
2015-08-15 16:00:24 - [CBlindLevels] Trying to find best matching bl 2015-08-15 16:00:24 - [CBlindLevels] Perfect match found 2015-08-15 16:00:24 - [CBlindLevels] Blinds recognized as 5.00 / 10. 2015-08-15 16:00:24 - [CSymbolEngineTableLimits] AutoLockBlinds() 2015-08-15 16:00:24 - [CSymbolEngineTableLimits] blinds_locked_for_c
```

Logging for DLL

Usually OpenHoldem will stop going into more detail when it reaches a DLL symbol, because DLL-functions and C-functions are a black-box for OpenHoldem. However it is possible to enable DLL-logging, then OpenHoldem will continue like in the axample above if the DLL-function calls back to OH-script. Furthermore it is possible to log any user-defined messages from the DLL to the standard OpenHoldem-log with a function from the DLL-interface:

```
EXE_IMPLEMENTS void __stdcall WriteLog(char* format, ...);
```

Now if your head hurts be assured: things are half as hard once you have gained a little bit experience.

The log-file knows the truth. True experts know the log-file.

Favourite Failures

This chapter will teach you about the most common beginners mistakes. They all have been reported at least several times; solving these "problems" over and over again has cost hundreds of newbies (and especially the forum-regulars!) quite some time and nerves. But the good news is: these issues are easy to avoid once you are aware of them. And the even better news: writing this manual took some time (due to all these newbies;-); in the mean-time we managed to prevent these beginner mistakes and to auto-detect most of the rest. There even is a new feature called "Problem Solver" that will assist you a little bit up to the point where OpenHoldem is able to act.



But it won't hurt if you read on and learn to know these potential pitfalls.

OpenHoldem does not connect to the table

Reason: Table not recognized because of incorrect system-settings. To recognize a table the following criteria have to match:

- size of the window
- table-title
- optional tablepoints r\$tablepointX if you have two very similar maps for the same casino, e.g. 6-max and 9-max.

Usually the problem is the window-size caused by "wrong" system-settings. Tablepoints are rarely used, but you should be aware of their existence, especially if you downloaded somebody elses map from the forum.

OpenHoldem does not play (I) and the autoplayerbutton can't be enabled

Reason: No good input.

The autoplayer of all more recent versions (4.x+) will immediately be enabled after connection to a table, but it can only act if the scraper recognizes your cards and at least two buttons.

OpenHoldem does not play (II)

Reason: no stable input ("balance-stability" and "card-stability")
One safety-measure to avoid incorrect plays is the option to wait for "balance-stability". Without reasonable input you can be sure to become a victim of the GIGO-principle (garbage in — garbage out). The example below shows a FullTilt mini-table, where the regions for player-names and balances partially overlap.

There are several counter-measures that will help you to deal with such issues:

- avoid mini-tables
- make sure, that regions don't overlap
- turn off all kind of animations
- *always use replay-frames* for building tablemaps and for investigation if you experience problems at a real casino.

OpenHoldem calls any hand

Reason: Tablemap does not detect bet amounts.

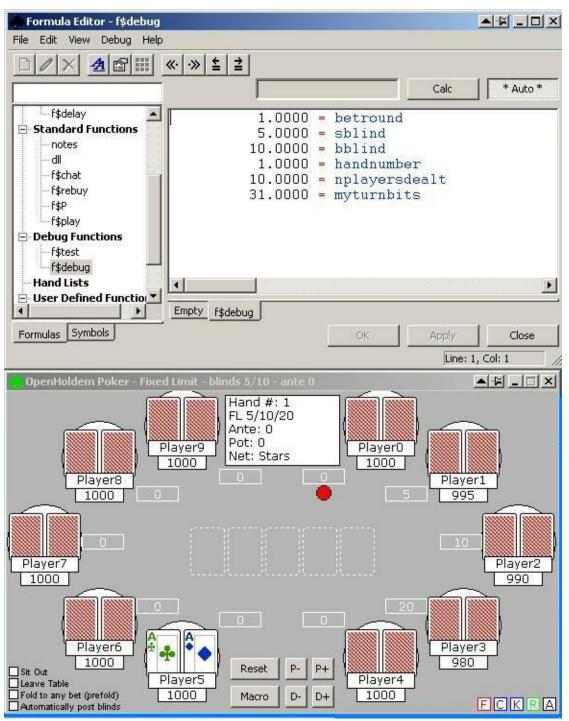
If OpenHoldem calls every hand — except maybe the strong ones — this is probably neither a bug in OpenHoldem nor bad logic in the bot, but a problem with the Tablemap or with your settings again: The TM does most probably not recognize your opponents bet, so OpenHoldem thinks you have a free play and calls.

Whenever you experience strange actions, you should ask yourself the following questions (http://forum.winholdem.net/wbb/viewtopic.php?f=38&t=12408):

- is the scraper reading the bet values correctly?
- is the scraper reading my cardfaces correctly?
- is the scraper reading opponent cardbacks correctly?
- do the blind values look sane?
- does the site provide handnumbers?
- are my action buttons being seen when my turn arrives?

You can check the values for these symbols in the debug-tab of the formulaeditor.

- = betround
- = sblind
- = bblind
- = handnumber
- = nplayersdealt
- = myturnbits



If you want to investigate in detail how OpenHoldem did choose a decision, you should lookup the full autoplayer-trace in the log-file.

OpenHoldem pushes any hand

Reason: can't connect to license server. ;-)

Just kidding — we don't have any license server. But in other forums we have

seen dialogues like below:

Customer: OpenHoldem pushes every hand.

(some days later): Any response? Admin: Our license server is down.

Customer: This can be the reason for allin?

Admin: Yes. One reason.

Customer: I bought early August. Now it is September.

Admin: Sorry. My partner is on holiday.

OpenHoldem does not work in ManualMode

Reason: Not your turn or userchair not recognized.

To work properly OpenHoldem must have recognized the userchair. Also some symbols are only valid when it is your turn to act. So you should highlight some buttons in ManualMode; just click them; usually they are grey, which means not existent.

OpenHoldem executes the wrong action

Sometimes new users complain that OpenHoldem executes the wrong action. The most common reason — after bad bot-logic — are "missing" buttons that did not get recognized by the tablemap. In the example above no raise-button could not be found that at many casinos gets used to confirm the betsize. As a consequence OpenHoldem chose a so-called backup-action.

OpenHoldem acts multiple times

Reason: Improper bring-delay and scraper-delay.

The bring-delay should be somewhat smaller than the scraper-delay. Otherwise it can happen that OpenHoldem acts but bring doesn't display a new image in time. So when the next heart-beat-cycle starts, OpenHoldem will scrape the exact same game-state another time and act another time.

This problem has been improved by marking frames as unstable when OpenHoldem acts, so OpenHoldem has to wait for a certain number of stable frames again. However the problem may still appear, if your scraper- and bringdelay do mismatch or if you don't wait for stable frames.

OpenHoldem permanently clicks sitout / sitin

Reason: Tablemap does not correctly scrape sitin/sitout.

The tablemap probably does not correctly determine the state of the sitin- or sitout-region; as a consequence the bot will think its action had no effect and will try over and over again. There is a funny legend back from the ancient times of WinHoldem or OPI: the tablemap (a built-in feature of the software at that time) once again failed because of a casino-update. As a consequence lots of "players" were suddenly sitting in and sitting out once per second. It must have been a devilish joy for their opponents; innummerable bots got reported and finally busted.

So always remember: Bots are very stupid and very volatile beings. This can no longer happen as we execute the hopper functions only once every 4 seconds.

OpenScrape does recognize something, but OpenHoldem does not (I)

Reason: Different versions of the same map.

A very common "bug"-report: some beginner has just fixed his first map and complains that OpenScrape recognizes the bets, the fonts of the player names or whatever; but OpenHoldem fails to do so. The reason is quite simple: he usually has multiple versions of the map in his scraper-folder and has simply loaded the wrong map. If you have the auto-connector enabled, OpenHoldem will load the "first" map that fits.

This gets now prevented as OpenHoldem checks if multiple maps with the same sitename exist.

OpenScrape does recognize something, but OpenHoldem does not (II)

Reason: Halfmade tablemap

Scraping is the most time-consuming part of the heartbeat-cycle, so OpenHoldem does implement some reasonable optimizations. It will only scrape names, bets and balances of a certain chair, if that chair is occupied. If OpenScrape and your tablemap are able to scrape those regions but OpenHoldem does not do so, then you have most probably forgotten to define pXseated or uXseated.

This gets now prevented, as OpenHoldem checks for complete tablemnaps.

OpenHoldem works locally, but not remotely with bring

Reason: Different system configurations on your casino-machine and your bot-machine.

It is a quite common beginners mistake to setup the casino-machine in a different way than the bot-machine. The casino-client will be rendered in a different way, so OpenHoldem will no longer be able to scrape the screenshots displayed by Bring. If you don't want to install OpenHoldem on the casino-machine to check the settings, the only way to fix that problem is a careful manual walk-through (experts could clone a VM or deploy some registry-settings).

OpenHoldem does not load my DLL (I).

Reason: Missing or incorrect libraries.

Most DLLs use external functionality and therefore depend on supporting libraries. Most often this is msvcrt (MicroSoft Visual Studio C++ RunTime Library). Also the correct version is important. OpenHoldem requires currently Visual Studio 2010 Professional, but Perl needs 2005. Checking the dependencies of your DLL with MicroSoft Dependency Walker will help. You can downlaod this tool from http://www.dependencywalker.com/.

OpenHoldem does not load my DLL (II).

Reason: Libraries not found.

Even if you have all libraries installed, it might be, that your operating system does not find your DLL or other required libraries. Windows uses the following "Dynamic-Link Library Search Order"

- 1. The directory from which the application loaded.
- 2. The current directory.
- 3. The system directory. Use the GetSystemDirectory function to get the path of this directory.
- 4. The 16-bit system directory. There is no function that obtains the path of this directory, but it is searched.
- 5. The Windows directory. Use the GetWindowsDirectory function to get the path of this directory.
- 6. The directories that are listed in the PATH environment variable. Note that this does not include the per-application path specified by the App Paths registry key. The App Paths key is not used when computing the DLL search path.

This is the default search order for Windows XP, SafeDllSearchMode disabled. It might be different for your operating system or configuration. For details please refer to http://msdn.microsoft.com/en-us/library/ms682586(VS.85).aspx.

OpenHoldem checked, but didfold is true

Reason: TM does not recognize buttons correctly.

If you get a freeplay, it might be that your check-button (or a check/fold-button!) is at the same location where usually the fold-button is located. If you have a simple way to detect buttons (e.g. by colour) you probably will recognize that button as fold. So OpenHoldem — not seeing a check-button — will check by clicking "fold" and didfold will be true. Similar things can happen if buttons move, e.g. a call-button at the raise-position when you can only fold or call allin.

History symbols do not work

Reason: Incorrect hand-reset.

Some symbols only require the current gamestate to be calculated (e.g. dealerchair, balance, nbetstocall). Others (like didcallround1, nbetsround3 etc.) require a complete game-history and will reset at the beginning of a new hand. If they all are zero (or -1), you have probably caused an incorrect hand-reset, e.g. by temporary occlusion. You should avoid occlusion by any means. This problem has now practically disappeared, since OpenHoldem supports ten different handreset-methods, but only accepts a hand-reset if multiple methods fire at the same time.

When the blinds increase OpenHoldem keeps the old values

Reason: tournament not recognized.

Detecting the blind values correctly is extremely important, as it affects all kinds of other symbols. Therefore OpenHoldem provides the ability to automatically lock the blinds as soon as they are known and reasonable. For the current hand and (for cashgames) for the rest of the session (after several hands have been played). As a consequence OpenHoldem needs to know if the current table belongs to a tournament and the blinds may go up. In the past there has been a tablemap-symbol r\$istournament, but way too much beginners (and veterans too!) mixed it up, so OpenHoldem detects tournaments now automatically based on several different factors;

- tournament identifiers in the title ("Sit&Go", "MTT", "satellite", "qualifier", ...)
- beautiful sum of chips at the table
- positive value of the symbol buyin (that can be read from the titlebar or scraped from the table)
- scraped MTT_symbols like players_remaining.

However in some very rare cases it might be that all methods of recognizing a tournament fail. Some casino tables don't display helpful titles or have no titles at all. The tablemap might not provide support for optional tournament information and even the "beautiful sum of chips" might not fit if you connect in the middle of an MTT. The methods of detecting tournaments still get fine-tuned, so if you experience this problem please get in contact with us. Istournament-help-threat:

http://www.maxinmontreal.com/forums/viewtopic.php?f=117&t=16104.

Pokerval does not work

Reason: Pokerval is a bitvector. You are probably confusing logical operations and bitwise operations.

In short: boolean operations work on true/false-values; bitwise operations work on long series of 0/1-values simultaneously. Lots of wisdom has been written about both topics — too much to explain it here. If you don't really understand what I am talking about, then you *really* should take a pause from coding and study the chapter about OH-script. There is no way around it.

PokerTracker does not work

Reason: Invalid Sitename in the TableMap

OpenHoldem supports Poker Tracker functionality for a dozen different casinos. To query the correct stats OpenHoldem must know, at what casino you currently play. This happens by the TableMap symbols *sitename* and *network*. Please refer to the PokerTracker and TableMap section of this manual for a detailed explanantion what to do.

Quickfolding AA and other unexpected plays

*Reason: Unstable frames and careless coding.*Consider code like below:

```
##list77##
AA KK QQ
AKs AQs
AKo
##f$prefold##
((betround == 1) AND NOT list77)
```

This code basically says: "do prefold, if is is preflop and I don't have a very great hand". However you only get stable frames when it is your turn. So it might be that you get a mis-scrape, e.g. occlusion by a rebuy-window. Or the poker-room might update its display, temporarily showing only one of your wonderful aces. So OpenHoldem can't recognize your great hand and folds. To avoid that problem, you need more fail-safe coding and explicitly specify the hands to fold: "do prefold, if it is preflop and I have a trash-hand."

```
##list13##
720 ....
##f$prefold##
((betround == 1) AND list13)
```

This way mis-scrapes will not be able to defeat your fine strategy. Instead a mis-scrape will only keep f\$prefold false and delay the folding of trash. The same problem might occur in all functions that might act on unstable frames (when it is not your turn), especially f\$prefold, f\$sitin, f\$sitout, f\$leave, f\$rebuy and f\$chat.

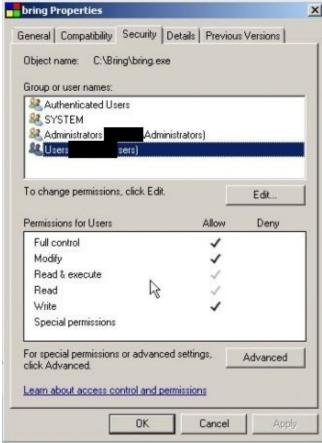
This failure gets now prevented too, as OpenHoldem will will execute f\$prefold only if both cards are known. However: as you can see lots of things could go wrong if you are careless. Therefore it is always a good idea to keep these potential pitfalls in mind and prefer a conservative style of coding your bot.

OpenHoldem acts, but nothing happens (Windows 7)

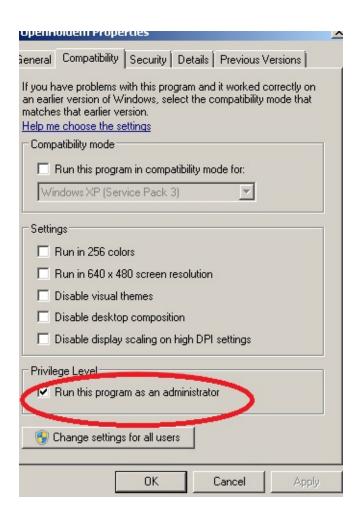
Some people have the problem that OpenHoldem "does not act" despite the log clearly shows the opposite:

```
2016-03-26 00:23:05 - Best action: f$betsize = 30.00 2016-03-26 00:23:05 - Action taken: f$betsize
```

The reason: OpenHoldem did act, but the casino did not react. As it turned out this problem was caused by Windows user account control (UAC), that prevents the generation of virtual keyboard events. OpenHoldem (or Bring) simply missed the necessary right to generate mouse-clicks and keyboard-events. To fix this just modify the permissions like below.

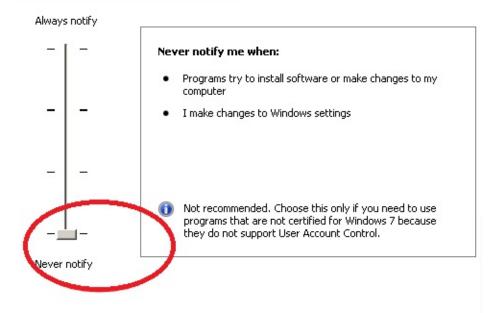


If all else fails run OpenHoldem as administrator.



Choose when to be notified about changes to your computer

User Account Control helps prevent potentially harmful programs from making changes to your computell me more about User Account Control settings





Installation and Setup

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-23T11:42:58.812000

Choosing an operating system

OpenHoldem has been originally designed to work on the 32-bit versions of Microsoft's Windows 2000 and Windows XP operating systems, but it supports all later versions too (and Windows 2000 no longer). When choosing a botting machine and an operating system the following points have to be considered:

- Does your favourite casino work at this system?
- How much resources especially RAM do the operating system and the casino need? (CPU is secondary, as at least OpenHoldem doesn't use much computing power. The author of this manual page could easily 250table at one of his machines if there were no crappy casinos and no casino security). Being leight-weight means that you could run multiple virtual machines at one hardware for better stealth and/or multiple casinos at the same time.
- Do you want to exchange tablemaps with other botters? (In the past most tablemaps were designed for Windows 2000 classic style, but since the introduction of automatic table-resizing most tablemaps work independently of the desktop theme.)
- Does your favourite casino support overlapping tables? The ability to take screenshots of partially or temporary hidden tables greatly improves the quality of the input and significantly reduces the risk of incorrect decisions if e.g. the table-hopper opens a new table that for a short moment overlaps another table that requires action.

The ability to scrape overlapping tables is only available at the more modern operating systems, but not at the leight-weight ones like Windows 2000 or Windows XP. So each operating system has its pros and cons and unfortunately there is no one that combines all the positive sides. So let's have a look at a concrete list.

- Windows 2000: completely outdated nowadays. MicroSoft ended support for this operating system and most casinos require more modern operating systems. Our compiler (MicroSoft Visual Studio 2015 Community Edition) does no longer create code for this system either, so OpenHoldem (the official version) no longer supports Windows 2000. If you really need it you would have to downgrade the project files and compile on your own.
- Windows XP: just like Windows 2000 this operating system is a bit old, but

most casinos still support it and and many botters prefer it for its stability and its small memory footprint. To reduce the memory usage even further many botters install minimalistic leight-weight versions (called e.g. "Tiny XP").

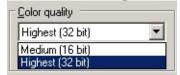
- *Windows Vista*: this operating system requires more resources than Windows XP, it needs at least 512 MB RAM (but this won't be enough, as the system alredy usess 448 MB minimum)..But Windows Vista provides a major advantage: scraping overlapping tables if the aero theme is enabled. No longer any bad input due to occluded tables is a major advantage for any screen-scraping poker-bot, that might out-weight the small disadvantages
- Windows 7: considered more stable as Windows Vista and according to a poll preferred by 40% of all botters nowadys. One common "problem" with Windows 7 however is its restrictice user account control (UAC). Most people finally chose to run OpenHoldem (and other botting software like hopper and Bring) as administrator.
- *Windows 8*: nearly unused in the botting-world, so we can't really assess this operating system.
- *Windows 10*: this is the most stable version of Windows we have ever seen and it supports overlapping tables out of the box. On the downside it needs at least 2 GB of RAM, but RAM is cheap nowadys, so Windows 10 became the preference of some experienced botters.

Configuring the operating system

OpenHoldem requires specific system settings in order to correctly intuit the game state from the pixels on-screen, and to correctly take poker actions. They have to be configured the right way before you can start working at your bot. Newer versions of OpenHoldem help you with this task; they execute automatical configuration checks and diagnose incorrect settings. However: you shouldn't rely completely on these configuration checks. Better get things right in advance. And be extra careful, if you use a two-computer-stealth-setup: then it is a must to setup both PCs in absolutely the same way.

Color depth

The display color depth must be set to 24-bit or 32-bit. Most Table Maps are designed for 32-bit color depth, so your display should be set to that for maximum compatibility with others' work.



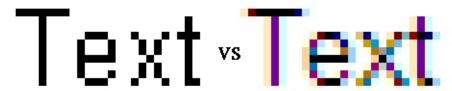
Right-Click on the desktop -> properties -> Settings

Display resolution

You need enough display resolution to contain all poker client windows. In general, the minimum acceptable display resolution is 1024x768. The smaller resolutions 800x600 and 640x480 do not have enough real estate to show both the poker client and OpenHoldem without overlapping windows.

Font smoothing

Font smoothing must be turned off in order to interpret font pixels correctly. Our objective is to interpret the pixels as presented by the poker client, and not post-mangled by the operating system. To disable font smoothing: right click the desktop, and choose Properties-->Appearance-->Effects. Disable smoothed fonts in this dialog.



Font Smoothing disabled / Font Smoothing enabled from Notepad

Window style (themes)

Windows 2000 has only one window style. There are no themes, there is no candy blue coloring around windows, so you are OK if you use Windows 2000. With Windows XP (or later), it is suggested to use the "Classic" theme, which will present windows in the same style as Windows 2000. You are of course free to use whatever theme you want, but be aware that public Table Maps usually do not work unless they are used with the same theme that they were designed for. Most OpenHoldem botters use Windows 2000 or the Windows XP Classic theme.



Windows XP and ClearType

If you use Windows XP, it is recommended that you disable ClearType (http://en.wikipedia.org/wiki/ClearType) fonts, for the same reason that we suggest you disable font smoothing. Here is the link to the Microsoft web page on how to do this:

http://www.microsoft.com/typography/cleartype/tuner/Step1.aspx.

Language settings

Your language settings affect the behaviour of your system in several ways:

- how to display special characters
- what to use as the decimal separator: dot or comma.
- the language used by your poker-client

Language settings may seem to be nearly irrelevant if you are a newbie, but

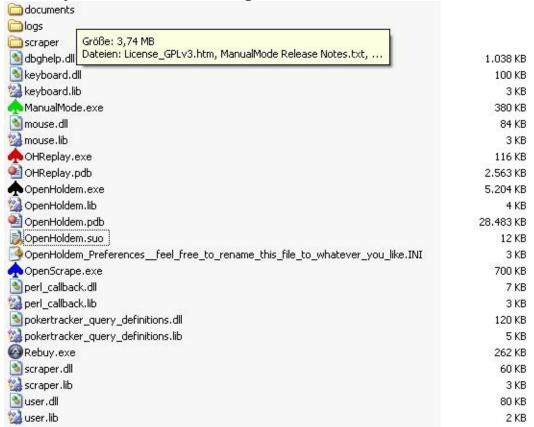
incorrect settings can affect the scraper engine and the autoplayer severly, even up to complete failure. The only settings, that are guaranteed to work are english text with US keyboard. Others may or may not be compatible. If you are new to botting be smart and stay on the safe side; there have been more than enough lengthy threads by desperate newbies — mostly caused by incorrect settings.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-10T00:27:23.147000

Installing the software

OpenHoldem and its supporting applications do not come with an installer. This is by design – installers tend to leave traces on your system, and in this hobby, the less traces, the better. Stealth will be discussed in greater depth in a later section, but this installer-less install is an important part of the stealth strategy. Since you do not have an installer to do the work for you, there are several steps you need to take to correctly configure the software on your machine:

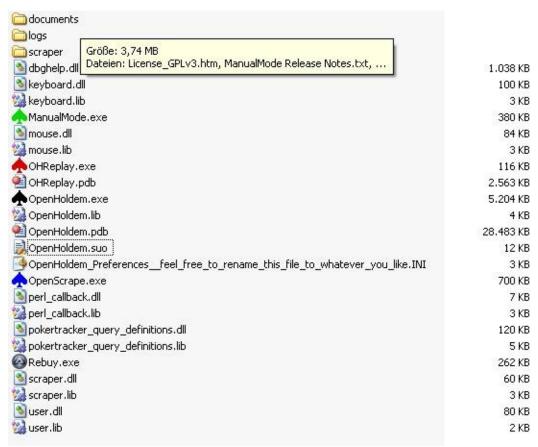
- 1. Create your "OpenHoldem" base directory. For purposes of this walkthrough, we will create this directory as "c:\mpb" ("My Poker Bot" clever, eh?). You can make this directory anything you want, however
- 2. Download the compiled binary packages. They can be found at the links in the {References} section
- 3. Extract all the files from the compiled binary packages into the c:\mpb directory, using WinRar, 7-zip or similar software. When done, your directory should look something like below.



4. OpenHoldem got created with MicroSoft Visual Studio 2015 Community

Edition and needs some extra MicroSoft DLLs (dynamic link libraries). Therefore you have to install the redistributable runtime-library from MicroSoft: https://www.microsoft.com/en-us/download/details.aspx? id=48145. You need the 32-bit-version, even if you have a 64-bit-system, because OpenHoldem still supports older machines with 32 bits.

OpenHoldem Directory



In the OpenHoldem directoryyou will find the following files and subdirectories:

- OpenHoldem.exe, ManualMode.exe, OHReplay.exe, Rebuy.exe these are the main files for each of the programs in the OpenHoldem project. Details on how to use each of these are throughout this manual.
- ______ this optional file is a large lookup table that is used by the versus symbols. As the versus database is quite large and does not change it is not included in the archives with recent executables. It is available at https://github.com/OpenHoldem/openholdembot/blob/master/versus.bin/vers10-13.rar?raw=true and should be extracted to the OpenHoldem directory.
- A file with .ini extensions this is the configuration file with your preferred settings; OpenHoldem won't store any information in the registry but use an ini-file instead; it can be named whatever you like, OpenHoldem will use it automatically, as long as there is only one ini-file in the directory.
- OpenHoldem_Manual.chm this manual.

- OpenPPL_Manual.chm the OpenPPL manual.
- pokertracker_query_definitions.dll a small DLL that contains the queries for all PokerTracker-symbols.
- symbols_pokertracker.pdf a short PDF-guide to the PokerTracker-symbols. This is a separate document, because the concept of query-definitions in a PokerTracker-DLL allows regular updates and fast additions of new PokerTracker-symbols.
- OpenPPL_Library.ohf a plain-text library that contains the definition of several 100 poker-logical symbols like StartingStacksize, NumberOfRaisesOnTurn, HadTopPairOnFlop, Position, etc. The OpenPPLlibrary will be loaded silently in the background and all symbols are automatically available both for OpenPPL and OH-script-users.
- \pm dbghelp.dll this is a Microsoft file that is used to produce memory dumps on critical errors. These memory dumps are useful for the developers to diagnose issues.
- ↓keyboard.dll, ↓mouse.dll, ↓scraper.dll, user.dll these files are all extensions to OpenHoldem that you can use to customize the software. More detail on all of these are provided in this document.
- *.lib-files these are linker files, necessary to link executables and DLLs against each other.
- Files with .pdb extensions these files contain the debug symbols that are required if you want to diagnose crashes or simply trace execution in a debugger. These files are usually not distributed with the executables as most end-users don't need them and the people who want to debug OpenHoldem are usually able to create debug-executables on their own.
- Documents-directory this directory contains useful information like the release-notes that document changes and the license-information (GPL v3). Feel free to peruse these if you have problems falling asleep some night.
- Logs this directory contains the log-files of the OpenHoldem bots. The logs are named "oh_0.log" .. "oh_24.log". The log-files provide very useful information if you want to track down mis-plays or even bugs. To get the correct file for a certain instance of OpenHoldem you can also use "Menu -> Edit -> View Log File" and OpenHoldem will open a text-editor with the correct log.
- Hand_Histories this directory contains auto-generated hand-histories for poker-sites that don't store hand-histories at your hard-drive.
- Scraper this directory contains the tablemaps for your favourite casinos.
 Just add your new tablemaps there once they have been tested;
 OpenHoldem will then load these tablemaps, so that its auto-connector will

be able to connect out-of-the box to any open table and start playing.

Finding a **↓**Table Map

Table maps provide the OpenHoldem game state engine the instructions for deriving the state of the game from the pixels on the screen. These files have the ".tm" extension on them and reside in the "scraper" directory where they get loaded automatically when OpenHoldem starts. The challenge with table maps is that every casino needs its own map. In some cases, you might need different maps for different game flavors. NL vs. FL, SNG vs. Ring, 10-seat vs. Heads-Up, for example.

Maintaining thousands of tablemaps would be a more than full time job, and thus instead we put the power in your hands; we provide Table Map creation tools and testing tools to allow you to be self-sufficient. If you favour a casino is not supported by any other bot or if your casino changes its display slightly you don't depend on anybody else. "OpenScrape" is the tool that helps you to adapt your tablemap or to create a completely new one if you find a nice new casino or game-type.

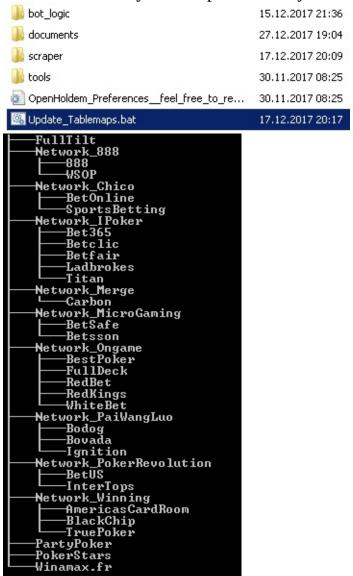
Tablemaps for simulators

Poker simulators are an easy starting point for every newbie and a good tool for experienced users to develop and test bot-logic offline. As an additional benefit their interface does not change, so their tablemaps don't need any updates OpenHoldem comes with a set of tablemaps for some offline simulators. . OpenHoldem ManualMode seems to work out of the box at every system, but some of the others need special settings: e.g. window size or screen resolution. Please see the Readme files for details.

- DDPoker, which is freeware and can be downloaded from http://www.ddpoker.com/
- OpenHoldem ManualMode, which is part of the OpenHoldem package
- PokerAcademy
- PokerSnowie
- PokerTH, which is open-source and can be downloaded from www.pokerth.net/

Downloading the "official" tablemaps

Since version 13 OpenHoldem provides a tablemap-distribution-service for "official" tablemaps. It requires the software: TortoiseSVN that you can get from https://tortoisesvn.net/downloads.html. Once TortoiseSVN is installed you simply execute the batch-script "Update_tablemaps.bat" and the latest tablemaps will be downloaded into your scraper-directory.



Please don't get fooled by this screenshot. The initial tablemap-distribution-service might contain some tablemaps less, but soon it will provide some dozen tablemaps for nearly all casinos and game-types. Our generous users make it happen.

Using the "official" tablemaps

All "official" tablemaps have been tested before release. The following setup is recommended if you are a beginner:

- any version of Windows should be OK, though the maps have been tested with Windows 7
- colour-depth of 24 or 32-bit is mandatory
- any screen-resolution should be OK, though the maps have been tested at 1024x768, as this resolution gets supported by nearly all machines.
- turning font-smoothing and clear-type fonts off is recommended. Some tablemaps use "fuzzy fonts" and accept some differences, but other tablemaps require an exact match.
- window-style (size of window-borders and titlebar) should no longer matter nowadays, as OpenHoldem cares only about the client-area of the table and good casinos do the same. If in doubt use the old Windows 2000 classic style.

Furthermore we recommend the following casino settings:

- default table-theme and background-colour, because we want you to configure as less as possible
- either default-size or mini-size (the tablemaps are able to resize the table, but first they have to connect to it)
- avatars turned off
- animatzions turned off

To ease the setup every real-world tablemap comes with two anonymised replayframes (call them "screen-shots") that illustrate the correct setup. One replayframe before connection and resizing, the other one after connection when it is heros turn postflop and some cards and the buttons are visible.

What if my casino is not yet supported?

If the "official" tablemaps do not yet support your favourite casino and gametype then chances are good that you will find a map in our forum at http://www.maxinmontreal.com/

	Forum	Topics	Posts
OpenHoldem			
<u>(a)</u>	OpenHoldem Table Map School Scraping and OCR forum Learn how to create and fix tablemaps Moderators: Ares®, Nick Coldhand, OpenMods	1646	11946
(2)	Free and working Tablemaps Created / fixed and contributed by our users Moderator: OpenMods	140	1478

Our users have posted several hundred tablemaps in the forum. Most probably your favourite casino and game-type is already covered. We simply did not yet have the time to test all these maps and make them "official". If you find a map that makes you happy you can help us:

- test it
- suggest it
- provide two replay-frames, one before connection and one when it is your turn postflop (shoot them with OpenReplayShooter; this tool creates HTML-files without any odentifying info, this makes anonymisation of the frames less work).

What if a tablemap does not work?

Don't worry. Fixing an existing map is often relative easy. It regularly comes down to adjusting a balance-region by some pixels or selecting a different colour for a font. These little fixes can be done by every dedicated newby. The chapter about tablemapping.explains the finer points in more depth and you will find some help in the tablemap school too.

↓Uninstallation

We hope you will never have to uninstall OpenHoldem on any of your machines. However, we often see enthusiastic newbies who download OpenHoldem and naively install it to their one and only machine without any attempts to stealth it. And then they ask how they can get their computer clean again — be it that they failed and want to play manually again or they want to create a clean setup for production.

Probably it is too late — you have already been flagged by your favourite casino. Anyway — the "best" way to get your PC "clean" again without a reinstallation:

- Remove the complete installation directory and all other botting-related files you may have on your computer: executables, formulae, tablemaps, archives, logs, etc.
- Delete any desktop icons and delete any startmenu shortcut (if you have created them) .
- OpenHoldem (version 4.x+) no longer uses the registry to store its preferences. However Windows stores some information like last recently used files. Take a registry-editor like regedit.exe and search for "openholdem", "openscrape", "manualmode", ".ohf", ".oppl", ".tm", "maxinmontreal" and maybe some others.
- Wipe your browser cache.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-09T23:19:45.038000

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. < http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License.

Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may

convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

• a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software

interchange.

- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be

resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part

of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the

covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General

Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE

COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or mo it under the terms of the GNU General Public License as publishe the Free Software Foundation, either version 3 of the License, o (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public Licens along with this program. If not, see http://www.gnu.org/licens

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html>.

Skip to main text Set language

Why GNU/Linux?

Search

English [en] Česky [cs] español [es] français [fr] italiano [it] [ja] [ko] português do Brasil [pt-br] русский [ru] polski [pl]



JOIN THE FSF

Sign up

Sponsored by the Free Software Foundation

About GNU Philosophy Licenses Education **Software Documentation Help GNU**

Frequently Asked Questions about the GNU Licenses

This page contains answers to commonly-asked questions about the GNU licenses.

This page is maintained by the Free Software Foundation's Licensing and Compliance Lab. You can support our efforts by making a donation to the FSF. Have a question not answered here? Check out some of our other licensing resources or contact the Compliance Lab at licensing@fsf.org.

Table of Contents

Basic questions about the GNU Project, the Free Software Foundation, and its licenses

- What does "GPL" stand for?
- Does free software mean using the GPL?
- Why should I use the GNU GPL rather than other free software licenses?
- <u>Does all GNU software use the GNU GPL as its license?</u>
- <u>Does using the GPL for a program make it GNU software?</u>
- Can I use the GPL for something other than software?
- Why don't you use the GPL for manuals?
- Are there translations of the GPL into other languages?
- Why are some GNU libraries released under the ordinary GPL rather than the Lesser GPL?
- Who has the power to enforce the GPL?
- Why does the FSF require that contributors to FSF-copyrighted
 programs assign copyright to the FSF? If I hold copyright on a GPL'ed

- program, should I do this, too? If so, how?
- Can I modify the GPL and make a modified license?
- Why did you decide to write the GNU Affero GPLv3 as a separate license?

General understanding of the GNU licenses

- Why does the GPL permit users to publish their modified versions?
- Does the GPL require that source code of modified versions be posted to the public?
- Can I have a GPL-covered program and an unrelated non-free program on the same computer?
- If I know someone has a copy of a GPL-covered program, can I demand he give me a copy?
- What does "written offer valid for any third party" mean in GPLv2?
 Does that mean everyone in the world can get the source to any
 GPL'ed program no matter what?
- The GPL says that modified versions, if released, must be "licensed ... to all third parties." Who are these third parties?
- Does the GPL allow me to sell copies of the program for money?
- Does the GPL allow me to charge a fee for downloading the program from my distribution site?
- Does the GPL allow me to require that anyone who receives the software must pay me a fee and/or notify me?
- If I distribute GPL'd software for a fee, am I required to also make it available to the public without a charge?
- <u>Does the GPL allow me to distribute a copy under a nondisclosure agreement?</u>
- Does the GPL allow me to distribute a modified or beta version under a nondisclosure agreement?
- Does the GPL allow me to develop a modified version under a nondisclosure agreement?
- Why does the GPL require including a copy of the GPL with every copy of the program?

What if the work is not very long?

Am I required to claim a copyright on my modifications to a GPL-covered program?

What does the GPL say about translating some code to a different programming language?

If a program combines public-domain code with GPL-covered code, can I take the public-domain part and use it as public domain code? I want to get credit for my work. I want people to know what I wrote. Can I still get credit if I use the GPL?

Does the GPL allow me to add terms that would require citation or acknowledgment in research papers which use the GPL-covered software or its output?

Can I omit the preamble of the GPL, or the instructions for how to use it on your own programs, to save space?

What does it mean to say that two licenses are "compatible"?

What does it mean to say a license is "compatible with the GPL"?

Why is the original BSD license incompatible with the GPL?

What is the difference between an "aggregate" and other kinds of "modified versions"?

Why does the FSF require that contributors to FSF-copyrighted programs assign copyright to the FSF? If I hold copyright on a GPL'ed program, should I do this, too? If so, how?

If I use a piece of software that has been obtained under the GNU GPL, am I allowed to modify the original code into a new program, then distribute and sell that new program commercially?

Can I use the GPL for something other than software?

I'd like to license my code under the GPL, but I'd also like to make it clear that it can't be used for military and/or commercial uses. Can I do this?

Can I use the GPL to license hardware?

<u>Does prelinking a GPLed binary to various libraries on the system, to optimize its performance, count as modification?</u>

How does the LGPL work with Java?

Why did you invent the new terms "propagate" and "convey" in

GPLv3?

- <u>Is "convey" in GPLv3</u> the same thing as what GPLv2 means by "distribute"?
- If I only make copies of a GPL-covered program and run them, without distributing or conveying them to others, what does the license require of me?
- GPLv3 gives "making available to the public" as an example of propagation. What does this mean? Is making available a form of conveying?
- Since distribution and making available to the public are forms of propagation that are also conveying in GPLv3, what are some examples of propagation that do not constitute conveying?
- How does GPLv3 make BitTorrent distribution easier?
- What is tivoization? How does GPLv3 prevent it?
- Does GPLv3 prohibit DRM?
- <u>Does GPLv3 require that voters be able to modify the software running in a voting machine?</u>
- Does GPLv3 have a "patent retaliation clause"?
- <u>In GPLv3 and AGPLv3</u>, what does it mean when it says "notwithstanding any other provision of this License"?
- <u>In AGPLv3</u>, what counts as "interacting with [the software] remotely through a computer network?"
- How does GPLv3's concept of "you" compare to the definition of "Legal Entity" in the Apache License 2.0?
- <u>In GPLv3</u>, what does "the <u>Program</u>" refer to? Is it every program ever released under <u>GPLv3</u>?
- <u>If some network client software is released under AGPLv3, does it have to be able to provide source to the servers it interacts with?</u>
- For software that runs a proxy server licensed under the AGPL, how can I provide an offer of source to users interacting with that code?

Using GNU licenses for your programs

■ How do I upgrade from (L)GPLv2 to (L)GPLv3?

- Could you give me step by step instructions on how to apply the GPL to my program?
- Why should I use the GNU GPL rather than other free software licenses?
- Why does the GPL require including a copy of the GPL with every copy of the program?
- <u>Is putting a copy of the GNU GPL in my repository enough to apply</u> the GPL?
- Why should I put a license notice in each source file?
- What if the work is not very long?
- Can I omit the preamble of the GPL, or the instructions for how to use it on your own programs, to save space?
- How do I get a copyright on my program in order to release it under the GPL?
- What if my school might want to make my program into its own proprietary software product?
- <u>I would like to release a program I wrote under the GNU GPL, but I would like to use the same code in non-free programs.</u>
- Can the developer of a program who distributed it under the GPL later license it to another party for exclusive use?
- Can the US Government release a program under the GNU GPL?
- Can the US Government release improvements to a GPL-covered program?
- Why should programs say "Version 3 of the GPL or any later version"?
- <u>Is it a good idea to use a license saying that a certain program can be used only under the latest version of the GNU GPL?</u>
- <u>Is there some way that I can GPL the output people get from use of my program? For example, if my program is used to develop hardware designs, can I require that these designs must be free?</u>
- Why don't you use the GPL for manuals?
- How does the GPL apply to fonts?
- What license should I use for website maintenance system templates?
- <u>Can I release a program under the GPL which I developed using non-free tools?</u>

- I use public key cryptography to sign my code to assure its authenticity. Is it true that GPLv3 forces me to release my private signing keys?
- <u>Does GPLv3 require that voters be able to modify the software running in a voting machine?</u>
- The warranty and liability disclaimers in GPLv3 seem specific to U.S. law. Can I add my own disclaimers to my own code?
- My program has interactive user interfaces that are non-visual in nature. How can I comply with the Appropriate Legal Notices requirement in GPLv3?

Distribution of programs released under the GNU licenses

- Can I release a modified version of a GPL-covered program in binary form only?
- <u>I downloaded just the binary from the net. If I distribute copies, do I have to get the source and distribute that too?</u>
- I want to distribute binaries via physical media without accompanying sources. Can I provide source code by FTP instead of by mail order?
- My friend got a GPL-covered binary with an offer to supply source, and made a copy for me. Can I use the offer to obtain the source?
- Can I put the binaries on my Internet server and put the source on a different Internet site?
- I want to distribute an extended version of a GPL-covered program in binary form. Is it enough to distribute the source for the original version?
- I want to distribute binaries, but distributing complete source is inconvenient. Is it ok if I give users the diffs from the "standard" version along with the binaries?
- Can I make binaries available on a network server, but send sources only to people who order them?
- How can I make sure each user who downloads the binaries also gets the source?

- Does the GPL require me to provide source code that can be built to match the exact hash of the binary I am distributing?
- Can I release a program with a license which says that you can distribute modified versions of it under the GPL but you can't distribute the original itself under the GPL?
- I just found out that a company has a copy of a GPL'ed program, and it costs money to get it. Aren't they violating the GPL by not making it available on the Internet?
- A company is running a modified version of a GPL'ed program on a web site. Does the GPL say they must release their modified sources?
- <u>Is use within one organization or company "distribution"?</u>
- <u>If someone steals a CD containing a version of a GPL-covered</u> <u>program, does the GPL give him the right to redistribute that version?</u>
- What if a company distributes a copy as a trade secret?
- Do I have "fair use" rights in using the source code of a GPL-covered program?
- Does moving a copy to a majority-owned, and controlled, subsidiary constitute distribution?
- Can software installers ask people to click to agree to the GPL? If I get some software under the GPL, do I have to agree to anything?
- I would like to bundle GPLed software with some sort of installation software. Does that installer need to have a GPL-compatible license?
- Does a distributor violate the GPL if they require me to "represent and warrant" that I am located in the US, or that I intend to distribute the software in compliance with relevant export control laws?
- The beginning of GPLv3 section 6 says that I can convey a covered work in object code form "under the terms of sections 4 and 5" provided I also meet the conditions of section 6. What does that mean?
- My company owns a lot of patents. Over the years we've contributed code to projects under "GPL version 2 or any later version", and the project itself has been distributed under the same terms. If a user decides to take the project's code (incorporating my contributions) under GPLv3, does that mean I've automatically granted GPLv3's explicit patent license to that user?

- <u>If I distribute a GPLv3-covered program, can I provide a warranty that is voided if the user modifies the program?</u>
- If I give a copy of a GPLv3-covered program to a coworker at my company, have I "conveyed" the copy to him?
- Am I complying with GPLv3 if I offer binaries on an FTP server and sources by way of a link to a source code repository in a version control system, like CVS or Subversion?
- Can someone who conveys GPLv3-covered software in a User Product use remote attestation to prevent a user from modifying that software?
- What does "rules and protocols for communication across the network" mean in GPLv3?
- Distributors that provide Installation Information under GPLv3 are not required to provide "support service" for the product. What kind of "support service" do you mean?

Using programs released under the GNU licenses when writing other programs

- Can I have a GPL-covered program and an unrelated non-free program on the same computer?
- Can I use GPL-covered editors such as GNU Emacs to develop non-free programs? Can I use GPL-covered tools such as GCC to compile them?
- Is there some way that I can GPL the output people get from use of my program? For example, if my program is used to develop hardware designs, can I require that these designs must be free?
- In what cases is the output of a GPL program covered by the GPL too?
- If I port my program to GNU/Linux, does that mean I have to release it as free software under the GPL or some other free software license?
- <u>I'd like to incorporate GPL-covered software in my proprietary system.</u>
 Can I do this?
- If I distribute a proprietary program that links against an LGPLv3-covered library that I've modified, what is the "contributor version" for purposes of determining the scope of the explicit patent license grant

- <u>I'm making—is it just the library, or is it the whole combination?</u>
- Under AGPLv3, when I modify the Program under section 13, what Corresponding Source does it have to offer?
- Where can I learn more about the GCC Runtime Library Exception?

Combining work with code released under the GNU licenses

- <u>Is GPLv3 compatible with GPLv2?</u>
- How are the various GNU licenses compatible with each other?
- What is the difference between an "aggregate" and other kinds of "modified versions"?
- <u>Do I have "fair use" rights in using the source code of a GPL-covered program?</u>
- Can the US Government release improvements to a GPL-covered program?
- Does the GPL have different requirements for statically vs dynamically linked modules with a covered work?
- Does the LGPL have different requirements for statically vs dynamically linked modules with a covered work?
- If a library is released under the GPL (not the LGPL), does that mean that any software which uses it has to be under the GPL or a GPL-compatible license?
- You have a GPL'ed program that I'd like to link with my code to build a proprietary program. Does the fact that I link with your program mean I have to GPL my program?
- If so, is there any chance I could get a license of your program under the Lesser GPL?
- <u>Using a certain GNU program under the GPL does not fit our project to make proprietary software. Will you make an exception for us? It would mean more users of that program.</u>
- If a programming language interpreter is released under the GPL, does that mean programs written to be interpreted by it must be under GPL-compatible licenses?

- <u>If a programming language interpreter has a license that is incompatible with the GPL, can I run GPL-covered programs on it?</u>
- If I add a module to a GPL-covered program, do I have to use the GPL as the license for my module?
- If I write a plug-in to use with a GPL-covered program, what requirements does that impose on the licenses I can use for distributing my plug-in?
- Can I apply the GPL when writing a plug-in for a non-free program?
- Can I release a non-free program that's designed to load a GPL-covered plug-in?
- <u>I'd like to incorporate GPL-covered software in my proprietary system.</u>
 Can I do this?
- I'd like to incorporate GPL-covered software in my proprietary system.

 Can I do this by putting a "wrapper" module, under a GPL-compatible lax permissive license (such as the X11 license) in between the GPL-covered part and the proprietary part?
- Can I write free software that uses non-free libraries?
- Can I link a GPL program with a proprietary system library?
- What legal issues come up if I use GPL-incompatible libraries with GPL software?
- I'm writing a Windows application with Microsoft Visual C++ and I will be releasing it under the GPL. Is dynamically linking my program with the Visual C++ runtime library permitted under the GPL?
- I'd like to modify GPL-covered programs and link them with the portability libraries from Money Guzzler Inc. I cannot distribute the source code for these libraries, so any user who wanted to change these versions would have to obtain those libraries separately. Why doesn't the GPL permit this?
- If license for a module Q has a requirement that's incompatible with the GPL, but the requirement applies only when Q is distributed by itself, not when Q is included in a larger program, does that make the license GPL-compatible? Can I combine or link Q with a GPL-covered program?
- In an object-oriented language such as Java, if I use a class that is

- GPL'ed without modifying, and subclass it, in what way does the GPL affect the larger program?
- Does distributing a nonfree driver meant to link with the kernel Linux violate the GPL?
- How can I allow linking of proprietary modules with my GPL-covered library under a controlled interface only?
- Consider this situation: 1) X releases V1 of a project under the GPL. 2) Y contributes to the development of V2 with changes and new code based on V1. 3) X wants to convert V2 to a non-GPL license. Does X need Y's permission?
- I have written an application that links with many different components, that have different licenses. I am very confused as to what licensing requirements are placed on my program. Can you please tell me what licenses I may use?
- Can I use snippets of GPL-covered source code within documentation that is licensed under some license that is incompatible with the GPL?

Questions about violations of the GNU licenses

- What should I do if I discover a possible violation of the GPL?
- Who has the power to enforce the GPL?
- <u>I heard that someone got a copy of a GPL'ed program under another</u> license. Is this possible?
- <u>Is the developer of a GPL-covered program bound by the GPL? Could</u> the developer's actions ever be a violation of the GPL?
- I just found out that a company has a copy of a GPL'ed program, and it costs money to get it. Aren't they violating the GPL by not making it available on the Internet?
- Can I use GPLed software on a device that will stop operating if customers do not continue paying a subscription fee?
- What does it mean to "cure" a violation of GPLv3?
- If someone installs GPLed software on a laptop, and then lends that laptop to a friend without providing source code for the software, have they violated the GPL?

Suppose that two companies try to circumvent the requirement to provide Installation Information by having one company release signed software, and the other release a User Product that only runs signed software from the first company. Is this a violation of GPLv3?

What does "GPL" stand for? (#WhatDoesGPLStandFor)

"GPL" stands for "General Public License". The most widespread such license is the GNU General Public License, or GNU GPL for short. This can be further shortened to "GPL", when it is understood that the GNU GPL is the one intended.

Does free software mean using the GPL?

(#DoesFreeSoftwareMeanUsingTheGPL)

Not at all—there are many other free software licenses. We have an <u>incomplete list</u>. Any license that provides the user <u>certain specific</u> freedoms is a free software license.

Why should I use the GNU GPL rather than other free software licenses? (#WhyUseGPL)

Using the GNU GPL will require that all the <u>released improved</u> <u>versions be free software</u>. This means you can avoid the risk of having to compete with a proprietary modified version of your own work. However, in some special situations it can be better to use a <u>more permissive license</u>.

Does all GNU software use the GNU GPL as its license?

(#DoesAllGNUSoftwareUseTheGNUGPLAsItsLicense)

Most GNU software packages use the GNU GPL, but there are a few GNU programs (and parts of programs) that use looser licenses,

such as the Lesser GPL. When we do this, it is a matter of <u>strategy</u>.

Does using the GPL for a program make it GNU software?

(#DoesUsingTheGPLForAProgramMakeItGNUSoftware)

Anyone can release a program under the GNU GPL, but that does not make it a GNU package.

Making the program a GNU software package means explicitly contributing to the GNU Project. This happens when the program's developers and the GNU Project agree to do it. If you are interested in contributing a program to the GNU Project, please write to maintainers@gnu.org.

What should I do if I discover a possible violation of the GPL? (#ReportingViolation)

You should <u>report it</u>. First, check the facts as best you can. Then tell the publisher or copyright holder of the specific GPL-covered program. If that is the Free Software Foundation, write to <u>violation@gnu.org</u>>. Otherwise, the program's maintainer may be the copyright holder, or else could tell you how to contact the copyright holder, so report it to the maintainer.

Why does the GPL permit users to publish their modified versions? (#WhyDoesTheGPLPermitUsersToPublishTheirModifiedVersions)

A crucial aspect of free software is that users are free to cooperate. It is absolutely essential to permit users who wish to help each other to share their bug fixes and improvements with other users.

Some have proposed alternatives to the GPL that require modified versions to go through the original author. As long as the original author keeps up with the need for maintenance, this may work well

in practice, but if the author stops (more or less) to do something else or does not attend to all the users' needs, this scheme falls down. Aside from the practical problems, this scheme does not allow users to help each other.

Sometimes control over modified versions is proposed as a means of preventing confusion between various versions made by users. In our experience, this confusion is not a major problem. Many versions of Emacs have been made outside the GNU Project, but users can tell them apart. The GPL requires the maker of a version to place his or her name on it, to distinguish it from other versions and to protect the reputations of other maintainers.

Does the GPL require that source code of modified versions be posted to the public? (#GPLRequireSourcePostedPublic)

The GPL does not require you to release your modified version, or any part of it. You are free to make modifications and use them privately, without ever releasing them. This applies to organizations (including companies), too; an organization can make a modified version and use it internally without ever releasing it outside the organization.

But *if* you release the modified version to the public in some way, the GPL requires you to make the modified source code available to the program's users, under the GPL.

Thus, the GPL gives permission to release the modified program in certain ways, and not in other ways; but the decision of whether to release it is up to you.

Can I have a GPL-covered program and an unrelated non-free program on the same computer? (#GPLAndNonfreeOnSameMachine)

If I know someone has a copy of a GPL-covered program, can I demand he give me a copy? (#CanIDemandACopy)

No. The GPL gives him permission to make and redistribute copies of the program *if and when he chooses to do so*. He also has the right not to redistribute the program, when that is what he chooses.

What does "written offer valid for any third party" mean in GPLv2? Does that mean everyone in the world can get the source to any GPL'ed program no matter what? (#WhatDoesWrittenOfferValid)

If you choose to provide source through a written offer, then anybody who requests the source from you is entitled to receive it.

If you commercially distribute binaries not accompanied with source code, the GPL says you must provide a written offer to distribute the source code later. When users non-commercially redistribute the binaries they received from you, they must pass along a copy of this written offer. This means that people who did not get the binaries directly from you can still receive copies of the source code, along with the written offer.

The reason we require the offer to be valid for any third party is so that people who receive the binaries indirectly in that way can order the source code from you.

GPLv2 says that modified versions, if released, must be "licensed ... to all third parties." Who are these third parties?

(#TheGPLSaysModifiedVersions)

Section 2 says that modified versions you distribute must be licensed to all third parties under the GPL. "All third parties" means

absolutely everyone—but this does not require you to *do* anything physically for them. It only means they have a license from you, under the GPL, for your version.

Am I required to claim a copyright on my modifications to a GPL-covered program? (#RequiredToClaimCopyright)

You are not required to claim a copyright on your changes. In most countries, however, that happens automatically by default, so you need to place your changes explicitly in the public domain if you do not want them to be copyrighted.

Whether you claim a copyright on your changes or not, either way you must release the modified version, as a whole, under the GPL (<u>if</u> you release your modified version at all).

What does the GPL say about translating some code to a different programming language? (#TranslateCode)

Under copyright law, translation of a work is considered a kind of modification. Therefore, what the GPL says about modified versions applies also to translated versions.

If a program combines public-domain code with GPL-covered code, can I take the public-domain part and use it as public domain code? (#CombinePublicDomainWithGPL)

You can do that, if you can figure out which part is the public domain part and separate it from the rest. If code was put in the public domain by its developer, it is in the public domain no matter where it has been.

Does the GPL allow me to sell copies of the program for money?

(#DoesTheGPLAllowMoney)

Yes, the GPL allows everyone to do this. The <u>right to sell copies</u> is part of the definition of free software. Except in one special situation, there is no limit on what price you can charge. (The one exception is the required written offer to provide source code that must accompany binary-only release.)

Does the GPL allow me to charge a fee for downloading the program from my distribution site?

Yes. You can charge any fee you wish for distributing a copy of the program. If you distribute binaries by download, you must provide "equivalent access" to download the source—therefore, the fee to download source may not be greater than the fee to download the binary.

Does the GPL allow me to require that anyone who receives the software must pay me a fee and/or notify me?

(#DoesTheGPLAllowRequireFee)

No. In fact, a requirement like that would make the program non-free. If people have to pay when they get a copy of a program, or if they have to notify anyone in particular, then the program is not free. See the <u>definition of free software</u>.

The GPL is a free software license, and therefore it permits people to use and even redistribute the software without being required to pay anyone a fee for doing so.

You *can* charge people a fee to <u>get a copy from you</u>. You can't require people to pay you when they get a copy *from someone else*.

If I distribute GPL'd software for a fee, am I required to also make

it available to the public without a charge?

(#DoesTheGPLRequireAvailabilityToPublic)

No. However, if someone pays your fee and gets a copy, the GPL gives them the freedom to release it to the public, with or without a fee. For example, someone could pay your fee, and then put her copy on a web site for the general public.

Does the GPL allow me to distribute copies under a nondisclosure agreement? (#DoesTheGPLAllowNDA)

No. The GPL says that anyone who receives a copy from you has the right to redistribute copies, modified or not. You are not allowed to distribute the work on any more restrictive basis.

If someone asks you to sign an NDA for receiving GPL-covered software copyrighted by the FSF, please inform us immediately by writing to license-violation@fsf.org.

If the violation involves GPL-covered code that has some other copyright holder, please inform that copyright holder, just as you would for any other kind of violation of the GPL.

Does the GPL allow me to distribute a modified or beta version under a nondisclosure agreement? (#DoesTheGPLAllowModNDA)

No. The GPL says that your modified versions must carry all the freedoms stated in the GPL. Thus, anyone who receives a copy of your version from you has the right to redistribute copies (modified or not) of that version. You may not distribute any version of the work on a more restrictive basis.

Does the GPL allow me to develop a modified version under a

nondisclosure agreement? (#DevelopChangesUnderNDA)

Yes. For instance, you can accept a contract to develop changes and agree not to release *your changes* until the client says ok. This is permitted because in this case no GPL-covered code is being distributed under an NDA.

You can also release your changes to the client under the GPL, but agree not to release them to anyone else unless the client says ok. In this case, too, no GPL-covered code is being distributed under an NDA, or under any additional restrictions.

The GPL would give the client the right to redistribute your version. In this scenario, the client will probably choose not to exercise that right, but does *have* the right.

I want to get credit for my work. I want people to know what I wrote. Can I still get credit if I use the GPL? (#IWantCredit)

You can certainly get credit for the work. Part of releasing a program under the GPL is writing a copyright notice in your own name (assuming you are the copyright holder). The GPL requires all copies to carry an appropriate copyright notice.

Does the GPL allow me to add terms that would require citation or acknowledgment in research papers which use the GPL-covered software or its output? (#RequireCitation)

No, this is not permitted under the terms of the GPL. While we recognize that proper citation is an important part academic publication, citation cannot be added as an additional requirement to the GPL. Requiring citation in research papers which made use of GPL'd software goes beyond what would be an acceptable additional requirement under section 7(b) of GPLv3, and therefore

would be considered an additional restriction under Section 7 of the GPL. And copyright law does not allow you to place such a requirement on the output of software, regardless of whether it is licensed under the terms of the GPL or some other license.

Why does the GPL require including a copy of the GPL with every copy of the program? (#WhyMustIInclude)

Including a copy of the license with the work is vital so that everyone who gets a copy of the program can know what his rights are.

It might be tempting to include a URL that refers to the license, instead of the license itself. But you cannot be sure that the URL will still be valid, five years or ten years from now. Twenty years from now, URLs as we know them today may no longer exist.

The only way to make sure that people who have copies of the program will continue to be able to see the license, despite all the changes that will happen in the network, is to include a copy of the license in the program.

Is it enough just to put a copy of the GNU GPL in my repository? (#LicenseCopyOnly)

Just putting a copy of the GNU GPL in a file in your repository does not explicitly state that the code in the same repository may be used under the GNU GPL. Without such a statement, it's not entirely clear that the permissions in the license really apply to any particular source file. An explicit statement saying that eliminates all doubt.

A file containing just a license, without a statement that certain other files are covered by that license, resembles a file containing just a subroutine which is never called from anywhere else. The resemblance is not perfect: lawyers and courts might apply common sense and conclude that you must have put the copy of the GNU GPL there because you wanted to license the code that way. But why leave any uncertainty?

This has nothing to do with the specifics of the GNU GPL. It is true for any free license.

Why should I put a license notice in each source file?

(#NoticeInSourceFile)

You should put a notice at the start of each source file, stating what license it carries, in order to avoid risk of the code's getting disconnected from its license. If your repository's README says that source file is under the GNU GPL, what happens if someone copies that file to another program? That other context may not show what the file's license is. It may appear to have some other license, or no license at all (which would make the code nonfree).

Adding a copyright notice and a license notice at the start of each source file is easy and makes such confusion unlikely.

This has nothing to do with the specifics of the GNU GPL. It is true for any free license.

What if the work is not very long? (#WhatIfWorkIsShort)

If a whole software package contains very little code—less than 300 lines is the benchmark we use—you may as well use a lax permissive license for it, rather than a copyleft license like the GNU GPL. (Unless, that is, the code is specially important.) We recommend the Apache License 2.0 for such cases.

Can I omit the preamble of the GPL, or the instructions for how to

use it on your own programs, to save space? (#GPLOmitPreamble)

The preamble and instructions are integral parts of the GNU GPL and may not be omitted. In fact, the GPL is copyrighted, and its license permits only verbatim copying of the entire GPL. (You can use the legal terms to make <u>another license</u> but it won't be the GNU GPL.)

The preamble and instructions add up to some 1000 words, less than 1/5 of the GPL's total size. They will not make a substantial fractional change in the size of a software package unless the package itself is quite small. In that case, you may as well use a simple all-permissive license rather than the GNU GPL.

What does it mean to say that two licenses are "compatible"? (#WhatIsCompatible)

In order to combine two programs (or substantial parts of them) into a larger work, you need to have permission to use both programs in this way. If the two programs' licenses permit this, they are compatible. If there is no way to satisfy both licenses at once, they are incompatible.

For some licenses, the way in which the combination is made may affect whether they are compatible—for instance, they may allow linking two modules together, but not allow merging their code into one module.

If you just want to install two separate programs in the same system, it is not necessary that their licenses be compatible, because this does not combine them into a larger work.

What does it mean to say a license is "compatible with the GPL?" (#WhatDoesCompatMean)

It means that the other license and the GNU GPL are compatible; you can combine code released under the other license with code released under the GNU GPL in one larger program.

All GNU GPL versions permit such combinations privately; they also permit distribution of such combinations provided the combination is released under the same GNU GPL version. The other license is compatible with the GPL if it permits this too.

GPLv3 is compatible with more licenses than GPLv2: it allows you to make combinations with code that has specific kinds of additional requirements that are not in GPLv3 itself. Section 7 has more information about this, including the list of additional requirements that are permitted.

Can I write free software that uses non-free libraries? (#FSWithNFLibs)

If you do this, your program won't be fully usable in a free environment. If your program depends on a non-free library to do a certain job, it cannot do that job in the Free World. If it depends on a non-free library to run at all, it cannot be part of a free operating system such as GNU; it is entirely off limits to the Free World.

So please consider: can you find a way to get the job done without using this library? Can you write a free replacement for that library?

If the program is already written using the non-free library, perhaps it is too late to change the decision. You may as well release the program as it stands, rather than not release it. But please mention in the README that the need for the non-free library is a drawback, and suggest the task of changing the program so that it does the same job without the non-free library. Please suggest that anyone who thinks of doing substantial further work on the program first free it from dependence on the non-free library.

Note that there may also be legal issues with combining certain non-free libraries with GPL-covered free software. Please see <u>the</u> <u>question on GPL software with GPL-incompatible libraries</u> for more information.

Can I link a GPL program with a proprietary system library? (#SystemLibraryException)

Both versions of the GPL have an exception to their copyleft, commonly called the system library exception. If the GPL-incompatible libraries you want to use meet the criteria for a system library, then you don't have to do anything special to use them; the requirement to distribute source code for the whole program does not include those libraries, even if you distribute a linked executable containing them.

The criteria for what counts as a "system library" vary between different versions of the GPL. GPLv3 explicitly defines "System Libraries" in section 1, to exclude it from the definition of "Corresponding Source." GPLv2 deals with this issue slightly differently, near the end of section 3.

What legal issues come up if I use GPL-incompatible libraries with GPL software? (#GPLIncompatibleLibs)

If you want your program to link against a library not covered by the system library exception, you need to provide permission to do that. Below are two example license notices that you can use to do that; one for GPLv3, and the other for GPLv2. In either case, you should put this text in each file to which you are granting this permission.

Only the copyright holders for the program can legally release their software under these terms. If you wrote the whole program yourself, then assuming your employer or school does not claim the

copyright, you are the copyright holder—so you can authorize the exception. But if you want to use parts of other GPL-covered programs by other authors in your code, you cannot authorize the exception for them. You have to get the approval of the copyright holders of those programs.

When other people modify the program, they do not have to make the same exception for their code—it is their choice whether to do so.

If the libraries you intend to link with are non-free, please also see the section on writing Free Software which uses non-free libraries.

If you're using GPLv3, you can accomplish this goal by granting an additional permission under section 7. The following license notice will do that. You must replace all the text in brackets with text that is appropriate for your program. If not everybody can distribute source for the libraries you intend to link with, you should remove the text in braces; otherwise, just remove the braces themselves.

Copyright (C) [years] [name of copyright holder]

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see http://www.gnu.org/licenses.

Additional permission under GNU GPL version 3 section 7

If you modify this Program, or any covered work, by linking or combining it with [name of library] (or a modified version of that library), containing parts covered by the terms of [name of library's license], the licensors of this Program grant you additional permission to convey the resulting work.

{Corresponding Source for a non-source form of such a combination shall include the source code for the parts of [name of library] used as well as that of the covered work.}

If you're using GPLv2, you can provide your own exception to the license's terms. The following license notice will do that. Again, you must replace all the text in brackets with text that is appropriate for your program. If not everybody can distribute source for the libraries you intend to link with, you should remove the text in braces; otherwise, just remove the braces themselves.

Copyright (C) [years] [name of copyright holder]

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see http://www.gnu.org/licenses.

Linking [name of your program] statically or dynamically with

other modules is making a combined work based on [name of your program]. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

In addition, as a special exception, the copyright holders of [name of your program] give you permission to combine [name of your program] with free software programs or libraries that are released under the GNU LGPL and with code included in the standard release of [name of library] under the [name of library's license] (or modified versions of such code, with unchanged license). You may copy and distribute such a system following the terms of the GNU GPL for [name of your program] and the licenses of the other code concerned{, provided that you include the source code of that other code when and as the GNU GPL requires distribution of source code}.

Note that people who make modified versions of [name of your program] are not obligated to grant this special exception for their modified versions; it is their choice whether to do so. The GNU General Public License gives permission to release a modified version without this exception; this exception also makes it possible to release a modified version which carries forward this exception.

How do I get a copyright on my program in order to release it under the GPL? (#HowIGetCopyright)

Under the Berne Convention, everything written is automatically copyrighted from whenever it is put in fixed form. So you don't have to do anything to "get" the copyright on what you write—as long as nobody else can claim to own your work.

However, registering the copyright in the US is a very good idea. It

will give you more clout in dealing with an infringer in the US.

The case when someone else might possibly claim the copyright is if you are an employee or student; then the employer or the school might claim you did the job for them and that the copyright belongs to them. Whether they would have a valid claim would depend on circumstances such as the laws of the place where you live, and on your employment contract and what sort of work you do. It is best to consult a lawyer if there is any possible doubt.

If you think that the employer or school might have a claim, you can resolve the problem clearly by getting a copyright disclaimer signed by a suitably authorized officer of the company or school. (Your immediate boss or a professor is usually NOT authorized to sign such a disclaimer.)

What if my school might want to make my program into its own proprietary software product? (#WhatIfSchool)

Many universities nowadays try to raise funds by restricting the use of the knowledge and information they develop, in effect behaving little different from commercial businesses. (See "The Kept University", Atlantic Monthly, March 2000, for a general discussion of this problem and its effects.)

If you see any chance that your school might refuse to allow your program to be released as free software, it is best to raise the issue at the earliest possible stage. The closer the program is to working usefully, the more temptation the administration might feel to take it from you and finish it without you. At an earlier stage, you have more leverage.

So we recommend that you approach them when the program is only half-done, saying, "If you will agree to releasing this as free software, I will finish it." Don't think of this as a bluff. To prevail, you must have the courage to say, "My program will have liberty, or never be born."

Could you give me step by step instructions on how to apply the GPL to my program? (#CouldYouHelpApplyGPL)

See the page of <u>GPL instructions</u>.

I heard that someone got a copy of a GPL'ed program under another license. Is this possible? (#HeardOtherLicense)

The GNU GPL does not give users permission to attach other licenses to the program. But the copyright holder for a program can release it under several different licenses in parallel. One of them may be the GNU GPL.

The license that comes in your copy, assuming it was put in by the copyright holder and that you got the copy legitimately, is the license that applies to your copy.

I would like to release a program I wrote under the GNU GPL, but I would like to use the same code in non-free programs.

(#ReleaseUnderGPLAndNF)

To release a non-free program is always ethically tainted, but legally there is no obstacle to your doing this. If you are the copyright holder for the code, you can release it under various different nonexclusive licenses at various times.

Is the developer of a GPL-covered program bound by the GPL? Could the developer's actions ever be a violation of the GPL?

(#DeveloperViolate)

Strictly speaking, the GPL is a license from the developer for others to use, distribute and change the program. The developer itself is not bound by it, so no matter what the developer does, this is not a "violation" of the GPL.

However, if the developer does something that would violate the GPL if done by someone else, the developer will surely lose moral standing in the community.

Can the developer of a program who distributed it under the GPL later license it to another party for exclusive use?

(#CanDeveloperThirdParty)

No, because the public already has the right to use the program under the GPL, and this right cannot be withdrawn.

Can I use GPL-covered editors such as GNU Emacs to develop non-free programs? Can I use GPL-covered tools such as GCC to compile them? (#CanIUseGPLToolsForNF)

Yes, because the copyright on the editors and tools does not cover the code you write. Using them does not place any restrictions, legally, on the license you use for your code.

Some programs copy parts of themselves into the output for technical reasons—for example, Bison copies a standard parser program into its output file. In such cases, the copied text in the output is covered by the same license that covers it in the source code. Meanwhile, the part of the output which is derived from the program's input inherits the copyright status of the input.

As it happens, Bison can also be used to develop non-free programs. This is because we decided to explicitly permit the use of the Bison standard parser program in Bison output files without restriction. We

made the decision because there were other tools comparable to Bison which already permitted use for non-free programs.

Do I have "fair use" rights in using the source code of a GPL-covered program? (#GPLFairUse)

Yes, you do. "Fair use" is use that is allowed without any special permission. Since you don't need the developers' permission for such use, you can do it regardless of what the developers said about it—in the license or elsewhere, whether that license be the GNU GPL or any other free software license.

Note, however, that there is no world-wide principle of fair use; what kinds of use are considered "fair" varies from country to country.

Can the US Government release a program under the GNU GPL? (#GPLUSGov)

If the program is written by US federal government employees in the course of their employment, it is in the public domain, which means it is not copyrighted. Since the GNU GPL is based on copyright, such a program cannot be released under the GNU GPL. (It can still be <u>free software</u>, however; a public domain program is free.)

However, when a US federal government agency uses contractors to develop software, that is a different situation. The contract can require the contractor to release it under the GNU GPL. (GNU Ada was developed in this way.) Or the contract can assign the copyright to the government agency, which can then release the software under the GNU GPL.

Can the US Government release improvements to a GPL-covered program? (#GPLUSGovAdd)

Yes. If the improvements are written by US government employees in the course of their employment, then the improvements are in the public domain. However, the improved version, as a whole, is still covered by the GNU GPL. There is no problem in this situation.

If the US government uses contractors to do the job, then the improvements themselves can be GPL-covered.

Does the GPL have different requirements for statically vs dynamically linked modules with a covered work?

(#GPLStaticVsDynamic)

No. Linking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination. See also What legal issues come up if I use GPL-incompatible libraries with GPL software?

Does the LGPL have different requirements for statically vs dynamically linked modules with a covered work?

(#LGPLStaticVsDynamic)

For the purpose of complying with the LGPL (any extant version: v2, v2.1 or v3):

(1) If you statically link against an LGPL'd library, you must also provide your application in an object (not necessarily source) format, so that a user has the opportunity to modify the library and relink the application.

(2) If you dynamically link against an LGPL'd library already present on the user's computer, you need not convey the library's source. On the other hand, if you yourself convey the executable LGPL'd library along with your application, whether linked with statically or dynamically, you must also convey the library's sources, in one of the ways for which the LGPL provides.

Is there some way that I can GPL the output people get from use of my program? For example, if my program is used to develop hardware designs, can I require that these designs must be free? (#GPLOutput)

In general this is legally impossible; copyright law does not give you any say in the use of the output people make from their data using your program. If the user uses your program to enter or convert his own data, the copyright on the output belongs to him, not you. More generally, when a program translates its input into some other form, the copyright status of the output inherits that of the input it was generated from.

So the only way you have a say in the use of the output is if substantial parts of the output are copied (more or less) from text in your program. For instance, part of the output of Bison (see above) would be covered by the GNU GPL, if we had not made an exception in this specific case.

You could artificially make a program copy certain text into its output even if there is no technical reason to do so. But if that copied text serves no practical purpose, the user could simply delete that text from the output and use only the rest. Then he would not have to obey the conditions on redistribution of the copied text.

In what cases is the output of a GPL program covered by the GPL

too? (#WhatCaseIsOutputGPL)

Only when the program copies part of itself into the output.

If I add a module to a GPL-covered program, do I have to use the GPL as the license for my module? (#GPLModuleLicense)

The GPL says that the whole combined program has to be released under the GPL. So your module has to be available for use under the GPL.

But you can give additional permission for the use of your code. You can, if you wish, release your program under a license which is more lax than the GPL but compatible with the GPL. The <u>license list page</u> gives a partial list of GPL-compatible licenses.

If a library is released under the GPL (not the LGPL), does that mean that any software which uses it has to be under the GPL or a GPL-compatible license? (#IfLibraryIsGPL)

Yes, because the software as it is actually run includes the library.

If a programming language interpreter is released under the GPL, does that mean programs written to be interpreted by it must be under GPL-compatible licenses? (#IfInterpreterIsGPL)

When the interpreter just interprets a language, the answer is no. The interpreted program, to the interpreter, is just data; a free software license like the GPL, based on copyright law, cannot limit what data you use the interpreter on. You can run it on any data (interpreted program), any way you like, and there are no requirements about licensing that data to anyone.

However, when the interpreter is extended to provide "bindings" to

other facilities (often, but not necessarily, libraries), the interpreted program is effectively linked to the facilities it uses through these bindings. So if these facilities are released under the GPL, the interpreted program that uses them must be released in a GPL-compatible way. The JNI or Java Native Interface is an example of such a binding mechanism; libraries that are accessed in this way are linked dynamically with the Java programs that call them. These libraries are also linked with the interpreter. If the interpreter is linked statically with these libraries, or if it is designed to link dynamically with these specific libraries, then it too needs to be released in a GPL-compatible way.

Another similar and very common case is to provide libraries with the interpreter which are themselves interpreted. For instance, Perl comes with many Perl modules, and a Java implementation comes with many Java classes. These libraries and the programs that call them are always dynamically linked together.

A consequence is that if you choose to use GPL'd Perl modules or Java classes in your program, you must release the program in a GPL-compatible way, regardless of the license used in the Perl or Java interpreter that the combined Perl or Java program will run on.

I'm writing a Windows application with Microsoft Visual C++ (or Visual Basic) and I will be releasing it under the GPL. Is dynamically linking my program with the Visual C++ (or Visual Basic) runtime library permitted under the GPL?

(#WindowsRuntimeAndGPL)

You may link your program to these libraries, and distribute the compiled program to others. When you do this, the runtime libraries are "System Libraries" as GPLv3 defines them. That means that you don't need to worry about including their source code with the program's Corresponding Source. GPLv2 provides a similar

exception in section 3.

You may not distribute these libraries in compiled DLL form with the program. To prevent unscrupulous distributors from trying to use the System Library exception as a loophole, the GPL says that libraries can only qualify as System Libraries as long as they're not distributed with the program itself. If you distribute the DLLs with the program, they won't be eligible for this exception anymore; then the only way to comply with the GPL would be to provide their source code, which you are unable to do.

It is possible to write free programs that only run on Windows, but it is not a good idea. These programs would be "<u>trapped</u>" by Windows, and therefore contribute zero to the Free World.

Why is the original BSD license incompatible with the GPL? (#OrigBSD)

Because it imposes a specific requirement that is not in the GPL; namely, the requirement on advertisements of the program. Section 6 of GPLv2 states:

You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

GPLv3 says something similar in section 10. The advertising clause provides just such a further restriction, and thus is GPL-incompatible.

The revised BSD license does not have the advertising clause, which eliminates the problem.

If I write a plug-in to use with a GPL-covered program, what requirements does that impose on the licenses I can use for

distributing my plug-in? (#GPLAndPlugins)

It depends on how the program invokes its plug-ins. If the program uses fork and exec to invoke plug-ins, then the plug-ins are separate programs, so the license for the main program makes no requirements for them.

If the program dynamically links plug-ins, and they make function calls to each other and share data structures, we believe they form a single program, which must be treated as an extension of both the main program and the plug-ins. This means you must license the plug-in under the GPL or a GPL-compatible free software license and distribute it with source code in a GPL-compliant way.

If the program dynamically links plug-ins, but the communication between them is limited to invoking the 'main' function of the plugin with some options and waiting for it to return, that is a borderline case.

Can I apply the GPL when writing a plug-in for a non-free program? (#GPLPluginsInNF)

If the program uses fork and exec to invoke plug-ins, then the plugins are separate programs, so the license for the main program makes no requirements for them. So you can use the GPL for a plugin, and there are no special requirements.

If the program dynamically links plug-ins, and they make function calls to each other and share data structures, we believe they form a single program, which must be treated as an extension of both the main program and the plug-ins. This means that combination of the GPL-covered plug-in with the non-free main program would violate the GPL. However, you can resolve that legal problem by adding an exception to your plug-in's license, giving permission to link it with the non-free main program.

See also the question <u>I am writing free software that uses a non-free library.</u>

Can I release a non-free program that's designed to load a GPL-covered plug-in? (#NFUseGPLPlugins)

It depends on how the program invokes its plug-ins. For instance, if the program uses *only* simple fork and exec to invoke and communicate with plug-ins, then the plug-ins are separate programs, so the license of the plug-in makes no requirements about the main program.

If the program dynamically links plug-ins, and they make function calls to each other and share data structures, we believe they form a single program, which must be treated as an extension of both the main program and the plug-ins. In order to use the GPL-covered plug-ins, the main program must be released under the GPL or a GPL-compatible free software license, and that the terms of the GPL must be followed when the main program is distributed for use with these plug-ins.

If the program dynamically links plug-ins, but the communication between them is limited to invoking the 'main' function of the plugin with some options and waiting for it to return, that is a borderline case.

Using shared memory to communicate with complex data structures is pretty much equivalent to dynamic linking.

See also the question <u>I am writing free software that uses a non-free library.</u>

You have a GPL'ed program that I'd like to link with my code to build a proprietary program. Does the fact that I link with your

program mean I have to GPL my program? (#LinkingWithGPL)

Not exactly. It means you must release your program under a license compatible with the GPL (more precisely, compatible with one or more GPL versions accepted by all the rest of the code in the combination that you link). The combination itself is then available under those GPL versions.

If so, is there any chance I could get a license of your program under the Lesser GPL? (#SwitchToLGPL)

You can ask, but most authors will stand firm and say no. The idea of the GPL is that if you want to include our code in your program, your program must also be free software. It is supposed to put pressure on you to release your program in a way that makes it part of our community.

You always have the legal alternative of not using our code.

Does distributing a nonfree driver meant to link with the kernel Linux violate the GPL? (#NonfreeDriverKernelLinux)

Linux (the kernel in the GNU/Linux operating system) is distributed under GNU GPL version 2. Does distributing a nonfree driver meant to link with Linux violate the GPL?

Yes, this is a violation, because effectively this makes a larger combined work. The fact that the user is expected to put the pieces together does not really change anything.

Each contributor to Linux who holds copyright on a substantial part of the code can enforce the GPL and we encourage each of them to take action against those distributing nonfree Linux-drivers.

How can I allow linking of proprietary modules with my GPL-covered library under a controlled interface only?

(#LinkingOverControlledInterface)

Add this text to the license notice of each file in the package, at the end of the text that says the file is distributed under the GNU GPL:

Linking ABC statically or dynamically with other modules is making a combined work based on ABC. Thus, the terms and conditions of the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of ABC give you permission to combine ABC program with free software programs or libraries that are released under the GNU LGPL and with independent modules that communicate with ABC solely through the ABCDEF interface. You may copy and distribute such a system following the terms of the GNU GPL for ABC and the licenses of the other code concerned, provided that you include the source code of that other code when and as the GNU GPL requires distribution of source code and provided that you do not modify the ABCDEF interface.

Note that people who make modified versions of ABC are not obligated to grant this special exception for their modified versions; it is their choice whether to do so. The GNU General Public License gives permission to release a modified version without this exception; this exception also makes it possible to release a modified version which carries forward this exception. If you modify the ABCDEF interface, this exception does not apply to your modified version of ABC, and you must remove this exception when you distribute your modified version.

This exception is an additional permission under section 7 of

the GNU General Public License, version 3 ("GPLv3")

This exception enables linking with differently licensed modules over the specified interface ("ABCDEF"), while ensuring that users would still receive source code as they normally would under the GPL.

Only the copyright holders for the program can legally authorize this exception. If you wrote the whole program yourself, then assuming your employer or school does not claim the copyright, you are the copyright holder—so you can authorize the exception. But if you want to use parts of other GPL-covered programs by other authors in your code, you cannot authorize the exception for them. You have to get the approval of the copyright holders of those programs.

I have written an application that links with many different components, that have different licenses. I am very confused as to what licensing requirements are placed on my program. Can you please tell me what licenses I may use? (#ManyDifferentLicenses)

To answer this question, we would need to see a list of each component that your program uses, the license of that component, and a brief (a few sentences for each should suffice) describing how your library uses that component. Two examples would be:

- To make my software work, it must be linked to the FOO library, which is available under the Lesser GPL.
- My software makes a system call (with a command line that I built) to run the BAR program, which is licensed under "the GPL, with a special exception allowing for linking with QUUX".

What is the difference between an "aggregate" and other kinds of

"modified versions"? (#MereAggregation)

An "aggregate" consists of a number of separate programs, distributed together on the same CD-ROM or other media. The GPL permits you to create and distribute an aggregate, even when the licenses of the other software are non-free or GPL-incompatible. The only condition is that you cannot release the aggregate under a license that prohibits users from exercising rights that each program's individual license would grant them.

Where's the line between two separate programs, and one program with two parts? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).

If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.

By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.

Why does the FSF require that contributors to FSF-copyrighted programs assign copyright to the FSF? If I hold copyright on a GPL'ed program, should I do this, too? If so, how? (#AssignCopyright)

Our lawyers have told us that to be in the best position to enforce

the GPL in court against violators, we should keep the copyright status of the program as simple as possible. We do this by asking each contributor to either assign the copyright on his contribution to the FSF, or disclaim copyright on it and thus put it in the public domain.

We also ask individual contributors to get copyright disclaimers from their employers (if any) so that we can be sure those employers won't claim to own the contributions.

Of course, if all the contributors put their code in the public domain, there is no copyright with which to enforce the GPL. So we encourage people to assign copyright on large code contributions, and only put small changes in the public domain.

If you want to make an effort to enforce the GPL on your program, it is probably a good idea for you to follow a similar policy. Please contact licensing@gnu.org if you want more information.

Can I modify the GPL and make a modified license? (#ModifyGPL)

It is possible to make modified versions of the GPL, but it tends to have practical consequences.

You can legally use the GPL terms (possibly modified) in another license provided that you call your license by another name and do not include the GPL preamble, and provided you modify the instructions-for-use at the end enough to make it clearly different in wording and not mention GNU (though the actual procedure you describe may be similar).

If you want to use our preamble in a modified license, please write to licensing@gnu.org for permission. For this purpose we would want to check the actual license requirements to see if we approve of them.

Although we will not raise legal objections to your making a modified license in this way, we hope you will think twice and not do it. Such a modified license is almost certainly <u>incompatible with the GNU GPL</u>, and that incompatibility blocks useful combinations of modules. The mere proliferation of different free software licenses is a burden in and of itself.

Rather than modifying the GPL, please use the exception mechanism offered by GPL version 3.

If I use a piece of software that has been obtained under the GNU GPL, am I allowed to modify the original code into a new program, then distribute and sell that new program commercially? (#GPLCommercially)

You are allowed to sell copies of the modified program commercially, but only under the terms of the GNU GPL. Thus, for instance, you must make the source code available to the users of the program as described in the GPL, and they must be allowed to redistribute and modify it as described in the GPL.

These requirements are the condition for including the GPL-covered code you received in a program of your own.

Can I use the GPL for something other than software?

(#GPLOtherThanSoftware)

You can apply the GPL to any kind of work, as long as it is clear what constitutes the "source code" for the work. The GPL defines this as the preferred form of the work for making changes in it.

However, for manuals and textbooks, or more generally any sort of work that is meant to teach a subject, we recommend using the GFDL rather than the GPL.

How does the LGPL work with Java? (#LGPLJava)

<u>See this article for details.</u> It works as designed, intended, and expected.

Consider this situation: 1) X releases V1 of a project under the GPL. 2) Y contributes to the development of V2 with changes and new code based on V1. 3) X wants to convert V2 to a non-GPL license. Does X need Y's permission? (#Consider)

Yes. Y was required to release its version under the GNU GPL, as a consequence of basing it on X's version V1. Nothing required Y to agree to any other license for its code. Therefore, X must get Y's permission before releasing that code under another license.

I'd like to incorporate GPL-covered software in my proprietary system. Can I do this? (#GPLInProprietarySystem)

You cannot incorporate GPL-covered software in a proprietary system. The goal of the GPL is to grant everyone the freedom to copy, redistribute, understand, and modify a program. If you could incorporate GPL-covered software into a non-free system, it would have the effect of making the GPL-covered software non-free too.

A system incorporating a GPL-covered program is an extended version of that program. The GPL says that any extended version of the program must be released under the GPL if it is released at all. This is for two reasons: to make sure that users who get the software get the freedom they should have, and to encourage people to give back improvements that they make.

However, in many cases you can distribute the GPL-covered software alongside your proprietary system. To do this validly, you must make sure that the free and non-free programs communicate at

arms length, that they are not combined in a way that would make them effectively a single program.

The difference between this and "incorporating" the GPL-covered software is partly a matter of substance and partly form. The substantive part is this: if the two programs are combined so that they become effectively two parts of one program, then you can't treat them as two separate programs. So the GPL has to cover the whole thing.

If the two programs remain well separated, like the compiler and the kernel, or like an editor and a shell, then you can treat them as two separate programs—but you have to do it properly. The issue is simply one of form: how you describe what you are doing. Why do we care about this? Because we want to make sure the users clearly understand the free status of the GPL-covered software in the collection.

If people were to distribute GPL-covered software calling it "part of" a system that users know is partly proprietary, users might be uncertain of their rights regarding the GPL-covered software. But if they know that what they have received is a free program plus another program, side by side, their rights will be clear.

I'd like to incorporate GPL-covered software in my proprietary system. Can I do this by putting a "wrapper" module, under a GPL-compatible lax permissive license (such as the X11 license) in between the GPL-covered part and the proprietary part? (#GPLWrapper)

No. The X11 license is compatible with the GPL, so you can add a module to the GPL-covered program and put it under the X11 license. But if you were to incorporate them both in a larger program, that whole would include the GPL-covered part, so it would have to be licensed *as a whole* under the GNU GPL.

The fact that proprietary module A communicates with GPL-covered module C only through X11-licensed module B is legally irrelevant; what matters is the fact that module C is included in the whole.

Where can I learn more about the GCC Runtime Library Exception? (#LibGCCException)

The GCC Runtime Library Exception covers libgcc, libstdc++, libfortran, libgomp, libdecnumber, and other libraries distributed with GCC. The exception is meant to allow people to distribute programs compiled with GCC under terms of their choice, even when parts of these libraries are included in the executable as part of the compilation process. To learn more, please read our <u>FAQ about the GCC Runtime Library Exception</u>.

I'd like to modify GPL-covered programs and link them with the portability libraries from Money Guzzler Inc. I cannot distribute the source code for these libraries, so any user who wanted to change these versions would have to obtain those libraries separately. Why doesn't the GPL permit this? (#MoneyGuzzlerInc)

There are two reasons for this. First, a general one. If we permitted company A to make a proprietary file, and company B to distribute GPL-covered software linked with that file, the effect would be to make a hole in the GPL big enough to drive a truck through. This would be carte blanche for withholding the source code for all sorts of modifications and extensions to GPL-covered software.

Giving all users access to the source code is one of our main goals, so this consequence is definitely something we want to avoid.

More concretely, the versions of the programs linked with the

Money Guzzler libraries would not really be free software as we understand the term—they would not come with full source code that enables users to change and recompile the program.

If the license for a module Q has a requirement that's incompatible with the GPL, but the requirement applies only when Q is distributed by itself, not when Q is included in a larger program, does that make the license GPL-compatible? Can I combine or link Q with a GPL-covered program? (#GPLIncompatibleAlone)

If a program P is released under the GPL that means *any and every part of it* can be used under the GPL. If you integrate module Q, and release the combined program P+Q under the GPL, that means any part of P+Q can be used under the GPL. One part of P+Q is Q. So releasing P+Q under the GPL says that Q any part of it can be used under the GPL. Putting it in other words, a user who obtains P+Q under the GPL can delete P, so that just Q remains, still under the GPL.

If the license of module Q permits you to give permission for that, then it is GPL-compatible. Otherwise, it is not GPL-compatible.

If the license for Q says in no uncertain terms that you must do certain things (not compatible with the GPL) when you redistribute Q on its own, then it does not permit you to distribute Q under the GPL. It follows that you can't release P+Q under the GPL either. So you cannot link or combine P with Q.

Can I release a modified version of a GPL-covered program in binary form only? (#ModifiedJustBinary)

No. The whole point of the GPL is that all modified versions must be <u>free software</u>—which means, in particular, that the source code of the modified version is available to the users.

I downloaded just the binary from the net. If I distribute copies, do I have to get the source and distribute that too? (#UnchangedJustBinary)

Yes. The general rule is, if you distribute binaries, you must distribute the complete corresponding source code too. The exception for the case where you received a written offer for source code is quite limited.

I want to distribute binaries via physical media without accompanying sources. Can I provide source code by FTP? (#DistributeWithSourceOnInternet)

Version 3 of the GPL allows this; see option 6(b) for the full details. Under version 2, you're certainly free to offer source via FTP, and most users will get it from there. However, if any of them would rather get the source on physical media by mail, you are required to provide that.

If you distribute binaries via FTP, <u>you should distribute source via FTP.</u>

My friend got a GPL-covered binary with an offer to supply source, and made a copy for me. Can I use the offer myself to obtain the source? (#RedistributedBinariesGetSource)

Yes, you can. The offer must be open to everyone who has a copy of the binary that it accompanies. This is why the GPL says your friend must give you a copy of the offer along with a copy of the binary—so you can take advantage of it.

Can I put the binaries on my Internet server and put the source on a different Internet site? (#SourceAndBinaryOnDifferentSites)

Yes. Section 6(d) allows this. However, you must provide clear instructions people can follow to obtain the source, and you must take care to make sure that the source remains available for as long as you distribute the object code.

I want to distribute an extended version of a GPL-covered program in binary form. Is it enough to distribute the source for the original version? (#DistributeExtendedBinary)

No, you must supply the source code that corresponds to the binary. Corresponding source means the source from which users can rebuild the same binary.

Part of the idea of free software is that users should have access to the source code for *the programs they use*. Those using your version should have access to the source code for your version.

A major goal of the GPL is to build up the Free World by making sure that improvement to a free program are themselves free. If you release an improved version of a GPL-covered program, you must release the improved source code under the GPL.

I want to distribute binaries, but distributing complete source is inconvenient. Is it ok if I give users the diffs from the "standard" version along with the binaries? (#DistributingSourceIsInconvenient)

This is a well-meaning request, but this method of providing the source doesn't really do the job.

A user that wants the source a year from now may be unable to get the proper version from another site at that time. The standard distribution site may have a newer version, but the same diffs probably won't work with that version. So you need to provide complete sources, not just diffs, with the binaries.

Can I make binaries available on a network server, but send sources only to people who order them? (#AnonFTPAndSendSources)

If you make object code available on a network server, you have to provide the Corresponding Source on a network server as well. The easiest way to do this would be to publish them on the same server, but if you'd like, you can alternatively provide instructions for getting the source from another server, or even a <u>version control system</u>. No matter what you do, the source should be just as easy to access as the object code, though. This is all specified in section 6(d) of GPLv3.

The sources you provide must correspond exactly to the binaries. In particular, you must make sure they are for the same version of the program—not an older version and not a newer version.

How can I make sure each user who downloads the binaries also gets the source? (#HowCanIMakeSureEachDownloadGetsSource)

You don't have to make sure of this. As long as you make the source and binaries available so that the users can see what's available and take what they want, you have done what is required of you. It is up to the user whether to download the source.

Our requirements for redistributors are intended to make sure the users can get the source code, not to force users to download the source code even if they don't want it.

Does the GPL require me to provide source code that can be built to match the exact hash of the binary I am distributing?

Complete corresponding source means the source that the binaries were made from, but that does not imply your tools must be able to make a binary that is an exact hash of the binary you are distributing. In some cases it could be (nearly) impossible to build a binary from source with an exact hash of the binary being distributed — consider the following examples: a system might put timestamps in binaries; or the program might have been built against a different (even unreleased) compiler version.

A company is running a modified version of a GPL'ed program on a web site. Does the GPL say they must release their modified sources? (#UnreleasedMods)

The GPL permits anyone to make a modified version and use it without ever distributing it to others. What this company is doing is a special case of that. Therefore, the company does not have to release the modified sources.

It is essential for people to have the freedom to make modifications and use them privately, without ever publishing those modifications. However, putting the program on a server machine for the public to talk to is hardly "private" use, so it would be legitimate to require release of the source code in that special case. Developers who wish to address this might want to use the <u>GNU Affero GPL</u> for programs designed for network server use.

Is making and using multiple copies within one organization or company "distribution"? (#InternalDistribution)

No, in that case the organization is just making the copies for itself. As a consequence, a company or other organization can develop a modified version and install that version through its own facilities,

without giving the staff permission to release that modified version to outsiders.

However, when the organization transfers copies to other organizations or individuals, that is distribution. In particular, providing copies to contractors for use off-site is distribution.

If someone steals a CD containing a version of a GPL-covered program, does the GPL give him the right to redistribute that version? (#StolenCopy)

If the version has been released elsewhere, then the thief probably does have the right to make copies and redistribute them under the GPL, but if he is imprisoned for stealing the CD he may have to wait until his release before doing so.

If the version in question is unpublished and considered by a company to be its trade secret, then publishing it may be a violation of trade secret law, depending on other circumstances. The GPL does not change that. If the company tried to release its version and still treat it as a trade secret, that would violate the GPL, but if the company hasn't released this version, no such violation has occurred.

What if a company distributes a copy as a trade secret? (#TradeSecretRelease)

If a company distributes a copy to you and claims it is a trade secret, the company has violated the GPL and will have to cease distribution. Note how this differs from the theft case above; the company does not intentionally distribute a copy when a copy is stolen, so in that case the company has not violated the GPL.

Why are some GNU libraries released under the ordinary GPL rather than the Lesser GPL? (#WhySomeGPLAndNotLGPL)

Using the Lesser GPL for any particular library constitutes a retreat for free software. It means we partially abandon the attempt to defend the users' freedom, and some of the requirements to share what is built on top of GPL-covered software. In themselves, those are changes for the worse.

Sometimes a localized retreat is a good strategy. Sometimes, using the LGPL for a library might lead to wider use of that library, and thus to more improvement for it, wider support for free software, and so on. This could be good for free software if it happens to a large extent. But how much will this happen? We can only speculate.

It would be nice to try out the LGPL on each library for a while, see whether it helps, and change back to the GPL if the LGPL didn't help. But this is not feasible. Once we use the LGPL for a particular library, changing back would be difficult.

So we decide which license to use for each library on a case-by-case basis. There is a <u>long explanation</u> of how we judge the question.

Using a certain GNU program under the GPL does not fit our project to make proprietary software. Will you make an exception for us? It would mean more users of that program.

(#WillYouMakeAnException)

Sorry, we don't make such exceptions. It would not be right.

Maximizing the number of users is not our aim. Rather, we are trying to give the crucial freedoms to as many users as possible. In general, proprietary software projects hinder rather than help the cause of freedom.

We do occasionally make license exceptions to assist a project which is producing free software under a license other than the GPL. However, we have to see a good reason why this will advance the cause of free software.

We also do sometimes change the distribution terms of a package, when that seems clearly the right way to serve the cause of free software; but we are very cautious about this, so you will have to show us very convincing reasons.

Why should programs say "Version 3 of the GPL or any later version"? (#VersionThreeOrLater)

From time to time, at intervals of years, we change the GPL—sometimes to clarify it, sometimes to permit certain kinds of use not previously permitted, and sometimes to tighten up a requirement. (The last two changes were in 2007 and 1991.) Using this "indirect pointer" in each program makes it possible for us to change the distribution terms on the entire collection of GNU software, when we update the GPL.

If each program lacked the indirect pointer, we would be forced to discuss the change at length with numerous copyright holders, which would be a virtual impossibility. In practice, the chance of having uniform distribution terms for GNU software would be nil.

Suppose a program says "Version 3 of the GPL or any later version" and a new version of the GPL is released. If the new GPL version gives additional permission, that permission will be available immediately to all the users of the program. But if the new GPL version has a tighter requirement, it will not restrict use of the current version of the program, because it can still be used under GPL version 3. When a program says "Version 3 of the GPL or any later version", users will always be permitted to use it, and even

change it, according to the terms of GPL version 3—even after later versions of the GPL are available.

If a tighter requirement in a new version of the GPL need not be obeyed for existing software, how is it useful? Once GPL version 4 is available, the developers of most GPL-covered programs will release subsequent versions of their programs specifying "Version 4 of the GPL or any later version". Then users will have to follow the tighter requirements in GPL version 4, for subsequent versions of the program.

However, developers are not obligated to do this; developers can continue allowing use of the previous version of the GPL, if that is their preference.

Is it a good idea to use a license saying that a certain program can be used only under the latest version of the GNU GPL? (#OnlyLatestVersion)

The reason you shouldn't do that is that it could result some day in withdrawing automatically some permissions that the users previously had.

Suppose a program was released in 2000 under "the latest GPL version". At that time, people could have used it under GPLv2. The day we published GPLv3 in 2007, everyone would have been suddenly compelled to use it under GPLv3 instead.

Some users may not even have known about GPL version 3—but they would have been required to use it. They could have violated the program's license unintentionally just because they did not get the news. That's a bad way to treat people.

We think it is wrong to take back permissions already granted, except due to a violation. If your freedom could be revoked, then it

isn't really freedom. Thus, if you get a copy of a program version under one version of a license, you should *always* have the rights granted by that version of the license. Releasing under "GPL version N or any later version" upholds that principle.

Why don't you use the GPL for manuals? (#WhyNotGPLForManuals)

It is possible to use the GPL for a manual, but the GNU Free Documentation License (GFDL) is much better for manuals.

The GPL was designed for programs; it contains lots of complex clauses that are crucial for programs, but that would be cumbersome and unnecessary for a book or manual. For instance, anyone publishing the book on paper would have to either include machine-readable "source code" of the book along with each printed copy, or provide a written offer to send the "source code" later.

Meanwhile, the GFDL has clauses that help publishers of free manuals make a profit from selling copies—cover texts, for instance. The special rules for Endorsements sections make it possible to use the GFDL for an official standard. This would permit modified versions, but they could not be labeled as "the standard".

Using the GFDL, we permit changes in the text of a manual that covers its technical topic. It is important to be able to change the technical parts, because people who change a program ought to change the documentation to correspond. The freedom to do this is an ethical imperative.

Our manuals also include sections that state our political position about free software. We mark these as "invariant", so that they cannot be changed or removed. The GFDL makes provisions for these "invariant sections".

How does the GPL apply to fonts? (#FontException)

Font licensing is a complex issue which needs serious consideration. The following license exception is experimental but approved for general use. We welcome suggestions on this subject—please see this this <u>explanatory essay</u> and write to <u>licensing@gnu.org</u>.

To use this exception, add this text to the license notice of each file in the package (to the extent possible), at the end of the text that says the file is distributed under the GNU GPL:

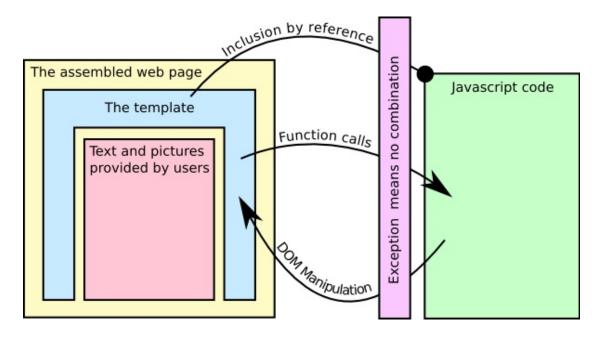
As a special exception, if you create a document which uses this font, and embed this font or unaltered portions of this font into the document, this font does not by itself cause the resulting document to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the document might be covered by the GNU General Public License. If you modify this font, you may extend this exception to your version of the font, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

I am writing a website maintenance system (called a "content management system" by some), or some other application which generates web pages from templates. What license should I use for those templates? (#WMS)

Templates are minor enough that it is not worth using copyleft to protect them. It is normally harmless to use copyleft on minor works, but templates are a special case, because they are combined with data provided by users of the application and the combination is distributed. So, we recommend that you license your templates under simple permissive terms.

Some templates make calls into JavaScript functions. Since

Javascript is often non-trivial, it is worth copylefting. Because the templates will be combined with user data, it's possible that template+user data+JavaScript would be considered one work under copyright law. A line needs to be drawn between the JavaScript (copylefted), and the user code (usually under incompatible terms).



Here's an exception for JavaScript code that does this:

As a special exception to the GPL, any HTML file which merely makes function calls to this code, and for that purpose includes it by reference shall be deemed a separate work for copyright law purposes. In addition, the copyright holders of this code give you permission to combine this code with free software libraries that are released under the GNU LGPL. You may copy and distribute such a system following the terms of the GNU GPL for this code and the LGPL for the libraries. If you modify this code, you may extend this exception to your version of the code, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Can I release a program under the GPL which I developed using

non-free tools? (#NonFreeTools)

Which programs you used to edit the source code, or to compile it, or study it, or record it, usually makes no difference for issues concerning the licensing of that source code.

However, if you link non-free libraries with the source code, that would be an issue you need to deal with. It does not preclude releasing the source code under the GPL, but if the libraries don't fit under the "system library" exception, you should affix an explicit notice giving permission to link your program with them. The FAQ entry about using GPL-incompatible libraries provides more information about how to do that.

Are there translations of the GPL into other languages? (#GPLTranslations)

It would be useful to have translations of the GPL into languages other than English. People have even written translations and sent them to us. But we have not dared to approve them as officially valid. That carries a risk so great we do not dare accept it.

A legal document is in some ways like a program. Translating it is like translating a program from one language and operating system to another. Only a lawyer skilled in both languages can do it—and even then, there is a risk of introducing a bug.

If we were to approve, officially, a translation of the GPL, we would be giving everyone permission to do whatever the translation says they can do. If it is a completely accurate translation, that is fine. But if there is an error in the translation, the results could be a disaster which we could not fix.

If a program has a bug, we can release a new version, and eventually the old version will more or less disappear. But once we have given everyone permission to act according to a particular translation, we have no way of taking back that permission if we find, later on, that it had a bug.

Helpful people sometimes offer to do the work of translation for us. If the problem were a matter of finding someone to do the work, this would solve it. But the actual problem is the risk of error, and offering to do the work does not avoid the risk. We could not possibly authorize a translation written by a non-lawyer.

Therefore, for the time being, we are not approving translations of the GPL as globally valid and binding. Instead, we are doing two things:

 Referring people to unofficial translations. This means that we permit people to write translations of the GPL, but we don't approve them as legally valid and binding.

An unapproved translation has no legal force, and it should say so explicitly. It should be marked as follows:

This translation of the GPL is informal, and not officially approved by the Free Software Foundation as valid. To be completely sure of what is permitted, refer to the original GPL (in English).

But the unapproved translation can serve as a hint for how to understand the English GPL. For many users, that is sufficient.

However, businesses using GNU software in commercial activity, and people doing public ftp distribution, should need to check the real English GPL to make sure of what it permits.

Publishing translations valid for a single country only.

We are considering the idea of publishing translations which are officially valid only for one country. This way, if there is a mistake, it will be limited to that country, and the damage will not be too great.

It will still take considerable expertise and effort from a sympathetic and capable lawyer to make a translation, so we cannot promise any such translations soon.

If a programming language interpreter has a license that is incompatible with the GPL, can I run GPL-covered programs on it? (#InterpreterIncompat)

When the interpreter just interprets a language, the answer is yes. The interpreted program, to the interpreter, is just data; the GPL doesn't restrict what tools you process the program with.

However, when the interpreter is extended to provide "bindings" to other facilities (often, but not necessarily, libraries), the interpreted program is effectively linked to the facilities it uses through these bindings. The JNI or Java Native Interface is an example of such a facility; libraries that are accessed in this way are linked dynamically with the Java programs that call them.

So if these facilities are released under a GPL-incompatible license, the situation is like linking in any other way with a GPL-incompatible library. Which implies that:

- 1. If you are writing code and releasing it under the GPL, you can state an explicit exception giving permission to link it with those GPL-incompatible facilities.
- 2. If you wrote and released the program under the GPL, and you designed it specifically to work with those facilities, people can take that as an implicit exception permitting them to link it with those facilities. But if that is what you

- intend, it is better to say so explicitly.
- 3. You can't take someone else's GPL-covered code and use it that way, or add such exceptions to it. Only the copyright holders of that code can add the exception.

Who has the power to enforce the GPL? (#WhoHasThePower)

Since the GPL is a copyright license, the copyright holders of the software are the ones who have the power to enforce the GPL. If you see a violation of the GPL, you should inform the developers of the GPL-covered software involved. They either are the copyright holders, or are connected with the copyright holders. <u>Learn more about reporting GPL violations</u>.

In an object-oriented language such as Java, if I use a class that is GPL'ed without modifying, and subclass it, in what way does the GPL affect the larger program? (#OOPLang)

Subclassing is creating a derivative work. Therefore, the terms of the GPL affect the whole program where you create a subclass of a GPL'ed class.

If I port my program to GNU/Linux, does that mean I have to release it as free software under the GPL or some other Free Software license? (#PortProgramToGL)

In general, the answer is no—this is not a legal requirement. In specific, the answer depends on which libraries you want to use and what their licenses are. Most system libraries either use the <u>GNU</u> <u>Lesser GPL</u>, or use the GNU GPL plus an exception permitting linking the library with anything. These libraries can be used in non-free programs; but in the case of the Lesser GPL, it does have some requirements you must follow.

Some libraries are released under the GNU GPL alone; you must use a GPL-compatible license to use those libraries. But these are normally the more specialized libraries, and you would not have had anything much like them on another platform, so you probably won't find yourself wanting to use these libraries for simple porting.

Of course, your software is not a contribution to our community if it is not free, and people who value their freedom will refuse to use it. Only people willing to give up their freedom will use your software, which means that it will effectively function as an inducement for people to lose their freedom.

If you hope some day to look back on your career and feel that it has contributed to the growth of a good and free society, you need to make your software free.

I just found out that a company has a copy of a GPL'ed program, and it costs money to get it. Aren't they violating the GPL by not making it available on the Internet? (#CompanyGPLCostsMoney)

No. The GPL does not require anyone to use the Internet for distribution. It also does not require anyone in particular to redistribute the program. And (outside of one special case), even if someone does decide to redistribute the program sometimes, the GPL doesn't say he has to distribute a copy to you in particular, or any other person in particular.

What the GPL requires is that he must have the freedom to distribute a copy to you *if he wishes to*. Once the copyright holder does distribute a copy of the program to someone, that someone can then redistribute the program to you, or to anyone else, as he sees fit.

Can I release a program with a license which says that you can distribute modified versions of it under the GPL but you can't

distribute the original itself under the GPL? (#ReleaseNotOriginal)

No. Such a license would be self-contradictory. Let's look at its implications for me as a user.

Suppose I start with the original version (call it version A), add some code (let's imagine it is 1000 lines), and release that modified version (call it B) under the GPL. The GPL says anyone can change version B again and release the result under the GPL. So I (or someone else) can delete those 1000 lines, producing version C which has the same code as version A but is under the GPL.

If you try to block that path, by saying explicitly in the license that I'm not allowed to reproduce something identical to version A under the GPL by deleting those lines from version B, in effect the license now says that I can't fully use version B in all the ways that the GPL permits. In other words, the license does not in fact allow a user to release a modified version such as B under the GPL.

Does moving a copy to a majority-owned, and controlled, subsidiary constitute distribution? (#DistributeSubsidiary)

Whether moving a copy to or from this subsidiary constitutes "distribution" is a matter to be decided in each case under the copyright law of the appropriate jurisdiction. The GPL does not and cannot override local laws. US copyright law is not entirely clear on the point, but appears not to consider this distribution.

If, in some country, this is considered distribution, and the subsidiary must receive the right to redistribute the program, that will not make a practical difference. The subsidiary is controlled by the parent company; rights or no rights, it won't redistribute the program unless the parent company decides to do so.

Can software installers ask people to click to agree to the GPL? If I get some software under the GPL, do I have to agree to anything? (#ClickThrough)

Some software packaging systems have a place which requires you to click through or otherwise indicate assent to the terms of the GPL. This is neither required nor forbidden. With or without a click through, the GPL's rules remain the same.

Merely agreeing to the GPL doesn't place any obligations on you. You are not required to agree to anything to merely use software which is licensed under the GPL. You only have obligations if you modify or distribute the software. If it really bothers you to click through the GPL, nothing stops you from hacking the GPLed software to bypass this.

I would like to bundle GPLed software with some sort of installation software. Does that installer need to have a GPL-compatible license? (#GPLCompatInstaller)

No. The installer and the files it installs are separate works. As a result, the terms of the GPL do not apply to the installation software.

Some distributors of GPL'd software require me in their umbrella EULAs or as part of their downloading process to "represent and warrant" that I am located in the US or that I intend to distribute the software in compliance with relevant export control laws. Why are they doing this and is it a violation of those distributors' obligations under GPL? (#ExportWarranties)

This is not a violation of the GPL. Those distributors (almost all of whom are commercial businesses selling free software distributions and related services) are trying to reduce their own legal risks, not to control your behavior. Export control law in the United States *might*

make them liable if they knowingly export software into certain countries, or if they give software to parties they know will make such exports. By asking for these statements from their customers and others to whom they distribute software, they protect themselves in the event they are later asked by regulatory authorities what they knew about where software they distributed was going to wind up. They are not restricting what you can do with the software, only preventing themselves from being blamed with respect to anything you do. Because they are not placing additional restrictions on the software, they do not violate section 10 of GPLv3 or section 6 of GPLv2.

The FSF opposes the application of US export control laws to free software. Not only are such laws incompatible with the general objective of software freedom, they achieve no reasonable governmental purpose, because free software is currently and should always be available from parties in almost every country, including countries that have no export control laws and which do not participate in US-led trade embargoes. Therefore, no country's government is actually deprived of free software by US export control laws, while no country's citizens should be deprived of free software, regardless of their governments' policies, as far as we are concerned. Copies of all GPL-licensed software published by the FSF can be obtained from us without making any representation about where you live or what you intend to do. At the same time, the FSF understands the desire of commercial distributors located in the US to comply with US laws. They have a right to choose to whom they distribute particular copies of free software; exercise of that right does not violate the GPL unless they add contractual restrictions beyond those permitted by the GPL.

Can I use GPLed software on a device that will stop operating if customers do not continue paying a subscription fee?

(#SubscriptionFee)

No. In this scenario, the requirement to keep paying a fee limits the user's ability to run the program. This is an additional requirement on top of the GPL, and the license prohibits it.

How do I upgrade from (L)GPLv2 to (L)GPLv3? (#v3HowToUpgrade)

First, include the new version of the license in your package. If you're using LGPLv3 in your project, be sure to include copies of both GPLv3 and LGPLv3, since LGPLv3 is now written as a set of additional permissions on top of GPLv3.

Second, replace all your existing v2 license notices (usually at the top of each file) with the new recommended text available on the <u>GNU licenses howto</u>. It's more future-proof because it no longer includes the FSF's postal mailing address.

Of course, any descriptive text (such as in a README) which talks about the package's license should also be updated appropriately.

How does GPLv3 make BitTorrent distribution easier? (#BitTorrent)

Because GPLv2 was written before peer-to-peer distribution of software was common, it is difficult to meet its requirements when you share code this way. The best way to make sure you are in compliance when distributing GPLv2 object code on BitTorrent would be to include all the corresponding source in the same torrent, which is prohibitively expensive.

GPLv3 addresses this problem in two ways. First, people who download this torrent and send the data to others as part of that process are not required to do anything. That's because section 9 says "Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance [of the license]."

Second, section 6(e) of GPLv3 is designed to give distributors—people who initially seed torrents—a clear and straightforward way to provide the source, by telling recipients where it is available on a public network server. This ensures that everyone who wants to get the source can do so, and it's almost no hassle for the distributor.

What is tivoization? How does GPLv3 prevent it? (#Tivoization)

Some devices utilize free software that can be upgraded, but are designed so that users are not allowed to modify that software. There are lots of different ways to do this; for example, sometimes the hardware checksums the software that is installed, and shuts down if it doesn't match an expected signature. The manufacturers comply with GPLv2 by giving you the source code, but you still don't have the freedom to modify the software you're using. We call this practice tivoization.

When people distribute User Products that include software under GPLv3, section 6 requires that they provide you with information necessary to modify that software. User Products is a term specially defined in the license; examples of User Products include portable music players, digital video recorders, and home security systems.

Does GPLv3 prohibit DRM? (#DRMProhibited)

It does not; you can use code released under GPLv3 to develop any kind of DRM technology you like. However, if you do this, section 3 says that the system will not count as an effective technological "protection" measure, which means that if someone breaks the DRM, he will be free to distribute his software too, unhindered by the DMCA and similar laws.

As usual, the GNU GPL does not restrict what people do in software, it just stops them from restricting others.

Can I use the GPL to license hardware? (#GPLHardware)

Any material that can be copyrighted can be licensed under the GPL. GPLv3 can also be used to license materials covered by other copyright-like laws, such as semiconductor masks. So, as an example, you can release a drawing of a physical object or circuit under the GPL.

In many situations, copyright does not cover making physical hardware from a drawing. In these situations, your license for the drawing simply can't exert any control over making or selling physical hardware, regardless of the license you use. When copyright does cover making hardware, for instance with IC masks, the GPL handles that case in a useful way.

I use public key cryptography to sign my code to assure its authenticity. Is it true that GPLv3 forces me to release my private signing keys? (#GiveUpKeys)

No. The only time you would be required to release signing keys is if you conveyed GPLed software inside a User Product, and its hardware checked the software for a valid cryptographic signature before it would function. In that specific case, you would be required to provide anyone who owned the device, on demand, with the key to sign and install modified software on his device so that it will run. If each instance of the device uses a different key, then you need only give each purchaser the key for his instance.

Does GPLv3 require that voters be able to modify the software running in a voting machine? (#v3VotingMachine)

No. Companies distributing devices that include software under GPLv3 are at most required to provide the source and Installation Information for the software to people who possess a copy of the

object code. The voter who uses a voting machine (like any other kiosk) doesn't get possession of it, not even temporarily, so the voter also does not get possession of the binary software in it.

Note, however, that voting is a very special case. Just because the software in a computer is free does not mean you can trust the computer for voting. We believe that computers cannot be trusted for voting. Voting should be done on paper.

Does GPLv3 have a "patent retaliation clause"? (#v3PatentRetaliation)

In effect, yes. Section 10 prohibits people who convey the software from filing patent suits against other licensees. If someone did so anyway, section 8 explains how they would lose their license and any patent licenses that accompanied it.

Can I use snippets of GPL-covered source code within documentation that is licensed under some license that is incompatible with the GPL? (#SourceCodeInDocumentation)

If the snippets are small enough that you can incorporate them under fair use or similar laws, then yes. Otherwise, no.

The beginning of GPLv3 section 6 says that I can convey a covered work in object code form "under the terms of sections 4 and 5" provided I also meet the conditions of section 6. What does that mean? (#v3Under4and5)

This means that all the permissions and conditions you have to convey source code also apply when you convey object code: you may charge a fee, you must keep copyright notices intact, and so on.

My company owns a lot of patents. Over the years we've

contributed code to projects under "GPL version 2 or any later version", and the project itself has been distributed under the same terms. If a user decides to take the project's code (incorporating my contributions) under GPLv3, does that mean I've automatically granted GPLv3's explicit patent license to that user? (#v2OrLaterPatentLicense)

No. When you convey GPLed software, you must follow the terms and conditions of one particular version of the license. When you do so, that version defines the obligations you have. If users may also elect to use later versions of the GPL, that's merely an additional permission they have—it does not require you to fulfill the terms of the later version of the GPL as well.

Do not take this to mean that you can threaten the community with your patents. In many countries, distributing software under GPLv2 provides recipients with an implicit patent license to exercise their rights under the GPL. Even if it didn't, anyone considering enforcing their patents aggressively is an enemy of the community, and we will defend ourselves against such an attack.

If I distribute a proprietary program that links against an LGPLv3-covered library that I've modified, what is the "contributor version" for purposes of determining the scope of the explicit patent license grant I'm making—is it just the library, or is it the whole combination? (#LGPLv3ContributorVersion)

The "contributor version" is only your version of the library.

Is GPLv3 compatible with GPLv2? (#v2v3Compatibility)

No. Some of the requirements in GPLv3, such as the requirement to provide Installation Information, do not exist in GPLv2. As a result, the licenses are not compatible: if you tried to combine code

released under both these licenses, you would violate section 6 of GPLv2.

However, if code is released under GPL "version 2 or later," that is compatible with GPLv3 because GPLv3 is one of the options it permits.

What does it mean to "cure" a violation of GPLv3? (#Cure)

To cure a violation means to adjust your practices to comply with the requirements of the license.

The warranty and liability disclaimers in GPLv3 seem specific to U.S. law. Can I add my own disclaimers to my own code? (#v3InternationalDisclaimers)

Yes. Section 7 gives you permission to add your own disclaimers, specifically 7(a).

My program has interactive user interfaces that are non-visual in nature. How can I comply with the Appropriate Legal Notices requirement in GPLv3? (#NonvisualLegalNotices)

All you need to do is ensure that the Appropriate Legal Notices are readily available to the user in your interface. For example, if you have written an audio interface, you could include a command that reads the notices aloud.

If I give a copy of a GPLv3-covered program to a coworker at my company, have I "conveyed" the copy to him? (#v3CoworkerConveying)

As long as you're both using the software in your work at the company, rather than personally, then the answer is no. The copies

belong to the company, not to you or the coworker. This copying is propagation, not conveying, because the company is not making copies available to others.

If I distribute a GPLv3-covered program, can I provide a warranty that is voided if the user modifies the program? (#v3ConditionalWarranty)

Yes. Just as devices do not need to be warranted if users modify the software inside them, you are not required to provide a warranty that covers all possible activities someone could undertake with GPLv3-covered software.

Why did you decide to write the GNU Affero GPLv3 as a separate license? (#SeparateAffero)

Early drafts of GPLv3 allowed licensors to add an Affero-like requirement to publish source in section 7. However, some companies that develop and rely upon free software consider this requirement to be too burdensome. They want to avoid code with this requirement, and expressed concern about the administrative costs of checking code for this additional requirement. By publishing the GNU Affero GPLv3 as a separate license, with provisions in it and GPLv3 to allow code under these licenses to link to each other, we accomplish all of our original goals while making it easier to determine which code has the source publication requirement.

Why did you invent the new terms "propagate" and "convey" in GPLv3? (#WhyPropagateAndConvey)

The term "distribute" used in GPLv2 was borrowed from United States copyright law. Over the years, we learned that some

jurisdictions used this same word in their own copyright laws, but gave it different meanings. We invented these new terms to make our intent as clear as possible no matter where the license is interpreted. They are not used in any copyright law in the world, and we provide their definitions directly in the license.

I'd like to license my code under the GPL, but I'd also like to make it clear that it can't be used for military and/or commercial uses. Can I do this? (#NoMilitary)

No, because those two goals contradict each other. The GNU GPL is designed specifically to prevent the addition of further restrictions. GPLv3 allows a very limited set of them, in section 7, but any other added restriction can be removed by the user.

Is "convey" in GPLv3 the same thing as what GPLv2 means by "distribute"? (#ConveyVsDistribute)

Yes, more or less. During the course of enforcing GPLv2, we learned that some jurisdictions used the word "distribute" in their own copyright laws, but gave it different meanings. We invented a new term to make our intent clear and avoid any problems that could be caused by these differences.

GPLv3 gives "making available to the public" as an example of propagation. What does this mean? Is making available a form of conveying? (#v3MakingAvailable)

One example of "making available to the public" is putting the software on a public web or FTP server. After you do this, some time may pass before anybody actually obtains the software from you—but because it could happen right away, you need to fulfill the GPL's obligations right away as well. Hence, we defined conveying

to include this activity.

Since distribution and making available to the public are forms of propagation that are also conveying in GPLv3, what are some examples of propagation that do not constitute conveying?

(#PropagationNotConveying)

Making copies of the software for yourself is the main form of propagation that is not conveying. You might do this to install the software on multiple computers, or to make backups.

Does prelinking a GPLed binary to various libraries on the system, to optimize its performance, count as modification? (#Prelinking)

No. Prelinking is part of a compilation process; it doesn't introduce any license requirements above and beyond what other aspects of compilation would. If you're allowed to link the program to the libraries at all, then it's fine to prelink with them as well. If you distribute prelinked object code, you need to follow the terms of section 6.

If someone installs GPLed software on a laptop, and then lends that laptop to a friend without providing source code for the software, have they violated the GPL? (#LaptopLoan)

No. In the jurisdictions where we have investigated this issue, this sort of loan would not count as conveying. The laptop's owner would not have any obligations under the GPL.

Suppose that two companies try to circumvent the requirement to provide Installation Information by having one company release signed software, and the other release a User Product that only runs signed software from the first company. Is this a violation of

GPLv3? (#TwoPartyTivoization)

Yes. If two parties try to work together to get around the requirements of the GPL, they can both be pursued for copyright infringement. This is especially true since the definition of convey explicitly includes activities that would make someone responsible for secondary infringement.

Am I complying with GPLv3 if I offer binaries on an FTP server and sources by way of a link to a source code repository in a version control system, like CVS or Subversion? (#SourceInCVS)

This is acceptable as long as the source checkout process does not become burdensome or otherwise restrictive. Anybody who can download your object code should also be able to check out source from your version control system, using a publicly available free software client. Users should be provided with clear and convenient instructions for how to get the source for the exact object code they downloaded—they may not necessarily want the latest development code, after all.

Can someone who conveys GPLv3-covered software in a User Product use remote attestation to prevent a user from modifying that software? (#RemoteAttestation)

No. The definition of Installation Information, which must be provided with source when the software is conveyed inside a User Product, explicitly says: "The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made." If the device uses remote attestation in some way, the Installation Information must provide you some means for your modified software to report itself as legitimate.

What does "rules and protocols for communication across the network" mean in GPLv3? (#RulesProtocols)

This refers to rules about traffic you can send over the network. For example, if there is a limit on the number of requests you can send to a server per day, or the size of a file you can upload somewhere, your access to those resources may be denied if you do not respect those limits.

These rules do not include anything that does not pertain directly to data traveling across the network. For instance, if a server on the network sent messages for users to your device, your access to the network could not be denied merely because you modified the software so that it did not display the messages.

Distributors that provide Installation Information under GPLv3 are not required to provide "support service" for the product. What kind of "support service" do you mean? (#SupportService)

This includes the kind of service many device manufacturers provide to help you install, use, or troubleshoot the product. If a device relies on access to web services or similar technology to function properly, those should normally still be available to modified versions, subject to the terms in section 6 regarding access to a network.

In GPLv3 and AGPLv3, what does it mean when it says "notwithstanding any other provision of this License"?

(#v3Notwithstanding)

This simply means that the following terms prevail over anything else in the license that may conflict with them. For example, without this text, some people might have claimed that you could not combine code under GPLv3 with code under AGPLv3, because the

AGPL's additional requirements would be classified as "further restrictions" under section 7 of GPLv3. This text makes clear that our intended interpretation is the correct one, and you can make the combination.

This text only resolves conflicts between different terms of the license. When there is no conflict between two conditions, then you must meet them both. These paragraphs don't grant you carte blanche to ignore the rest of the license—instead they're carving out very limited exceptions.

Under AGPLv3, when I modify the Program under section 13, what Corresponding Source does it have to offer?

(#AGPLv3CorrespondingSource)

"Corresponding Source" is defined in section 1 of the license, and you should provide what it lists. So, if your modified version depends on libraries under other licenses, such as the Expat license or GPLv3, the Corresponding Source should include those libraries (unless they are System Libraries). If you have modified those libraries, you must provide your modified source code for them.

The last sentence of the first paragraph of section 13 is only meant to reinforce what most people would have naturally assumed: even though combinations with code under GPLv3 are handled through a special exception in section 13, the Corresponding Source should still include the code that is combined with the Program this way. This sentence does not mean that you *only* have to provide the source that's covered under GPLv3; instead it means that such code is *not* excluded from the definition of Corresponding Source.

In AGPLv3, what counts as "interacting with [the software] remotely through a computer network?" (#AGPLv3InteractingRemotely)

If the program is expressly designed to accept user requests and send responses over a network, then it meets these criteria. Common examples of programs that would fall into this category include web and mail servers, interactive web-based applications, and servers for games that are played online.

If a program is not expressly designed to interact with a user through a network, but is being run in an environment where it happens to do so, then it does not fall into this category. For example, an application is not required to provide source merely because the user is running it over SSH, or a remote X session.

How does GPLv3's concept of "you" compare to the definition of "Legal Entity" in the Apache License 2.0? (#ApacheLegalEntity)

They're effectively identical. The definition of "Legal Entity" in the Apache License 2.0 is very standard in various kinds of legal agreements—so much so that it would be very surprising if a court did not interpret the term in the same way in the absence of an explicit definition. We fully expect them to do the same when they look at GPLv3 and consider who qualifies as a licensee.

In GPLv3, what does "the Program" refer to? Is it every program ever released under GPLv3? (#v3TheProgram)

The term "the Program" means one particular work that is licensed under GPLv3 and is received by a particular licensee from an upstream licensor or distributor. The Program is the particular work of software that you received in a given instance of GPLv3 licensing, as you received it.

"The Program" cannot mean "all the works ever licensed under GPLv3"; that interpretation makes no sense for a number of reasons. We've published an <u>analysis of the term "the Program"</u> for those

who would like to learn more about this.

If I only make copies of a GPL-covered program and run them, without distributing or conveying them to others, what does the license require of me? (#NoDistributionRequirements)

Nothing. The GPL does not place any conditions on this activity.

If some network client software is released under AGPLv3, does it have to be able to provide source to the servers it interacts with? (#AGPLv3ServerAsUser)

This should not be required in any typical server-client relationship. AGPLv3 requires a program to offer source code to "all users interacting with it remotely through a computer network." In most server-client architectures, it simply wouldn't be reasonable to argue that the server operator is a "user" interacting with the client in any meaningful sense.

Consider HTTP as an example. All HTTP clients expect servers to provide certain functionality: they should send specified responses to well-formed requests. The reverse is not true: servers cannot assume that the client will do anything in particular with the data they send. The client may be a web browser, an RSS reader, a spider, a network monitoring tool, or some special-purpose program. The server can make absolutely no assumptions about what the client will do—so there's no meaningful way for the server operator to be considered a user of that software.

For software that runs a proxy server licensed under the AGPL, how can I provide an offer of source to users interacting with that code? (#AGPLProxy)

For software on a proxy server, you can provide an offer of source through a normal method of delivering messages to users of that kind of proxy. For example, a Web proxy could use a landing page. When users initially start using the proxy, you can direct them to a page with the offer of source along with any other information you choose to provide.

The AGPL says you must make the offer to "all users". If you know that a certain user has already been shown the offer, for the current version of the software, you don't have to repeat it to that user again.

How are the various GNU licenses compatible with each other? (#AllCompatibility)

The various GNU licenses enjoy broad compatibility between each other. The only time you may not be able to combine code under two of these licenses is when you want to use code that's *only* under an older version of a license with code that's under a newer version.

Below is a detailed compatibility matrix for various combinations of the GNU licenses, to provide an easy-to-use reference for specific cases. It assumes that someone else has written some software under one of these licenses, and you want to somehow incorporate code from that into a project that you're releasing (either your own original work, or a modified version of someone else's software). Find the license for your project in a column at the top of the table, and the license for the other code in a row on the left. The cell where they meet will tell you whether or not this combination is permitted.

When we say "copy code," we mean just that: you're taking a section of code from one source, with or without modification, and inserting it into your own program, thus forming a work based on the first section of code. "Use a library" means that you're not copying any source directly, but instead interacting with it through

linking, importing, or other typical mechanisms that bind the sources together when you compile or run the code.

Skip compatibility matrix

			I want to license my		
		GPLv2 only	GPLv2 or later	GPLv3 or later	L
	GPLv2 only	OK	OK [2]	NO	Co
	GPLv2 or later	OK [<u>1</u>]	OK	OK	Co j C
I want to copy	GPLv3	NO	OK: Combination is under GPLv3 [3]	OK	Co G
code under:	LGPLv2.1 only	OK: Convey copied code under GPLv2 [7]	OK: Convey copied code under GPLv2 or later [7]	OK: Convey copied code under GPLv3 [7]	
	LGPLv2.1 or later	OK: Convey copied code under GPLv2 [7][1]	OK: Convey copied code under GPLv2 or later [7]	OK: Convey code under GPLv3 [7]	
	LGPLv3	NO	OK: Combination is under GPLv3 [8][3]	OK: Combination is under GPLv3 [8]	Co GP

	GPLv2 only	OK	OK [<u>2</u>]	NO	Co
I want	GPLv2 or later	OK [1]	OK	OK	Co j C
to use a library under:	GPLv3	NO	OK: Combination is under GPLv3 [3]	OK	Co G
	LGPLv2.1 only	OK	OK OK	OK	
	LGPLv2.1 or later	OK	OK	OK	
	LGPLv3	NO	OK: Combination is under GPLv3 [9]	OK	

Skip footnotes

- 1: You must follow the terms of GPLv2 when incorporating the code in this case. You cannot take advantage of terms in later versions of the GPL.
- 2: While you may release your project (either your original work and/or work that you received and modified) under GPLv2-or-later in this case, note that the other code you're using must remain under GPLv2 only. As long as your project depends on that code, you won't be able to upgrade the license of your project to GPLv3-or-

later, and the work as a whole (any combination of both your project and the other code) can only be conveyed under the terms of GPLv2.

- 3: If you have the ability to release the project under GPLv2 or any later version, you can choose to release it under GPLv3 or any later version—and once you do that, you'll be able to incorporate the code released under GPLv3.
- 4: If you have the ability to release the project under LGPLv2.1 or any later version, you can choose to release it under LGPLv3 or any later version—and once you do that, you'll be able to incorporate the code released under LGPLv3.
- 5: You must follow the terms of LGPLv2.1 when incorporating the code in this case. You cannot take advantage of terms in later versions of the LGPL.
- 6: If you do this, as long as the project contains the code released under LGPLv2.1 only, you will not be able to upgrade the project's license to LGPLv3 or later.
- 7: LGPLv2.1 gives you permission to relicense the code under any version of the GPL since GPLv2. If you can switch the LGPLed code in this case to using an appropriate version of the GPL instead (as noted in the table), you can make this combination.
- 8: LGPLv3 is GPLv3 plus extra permissions that you can ignore in this case.
- 9: Because GPLv2 does not permit combinations with LGPLv3, you must convey the project under GPLv3's terms in this case, since it will allow that combination.

GNU home page FSF home page GNU Art GNU Fun GNU's Who? Free Software Directory Site map



"Our mission is to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software users."

The <u>Free Software Foundation</u> is the principal organizational sponsor of the GNU Operating System. **Support GNU and the FSF** by <u>buying manuals and gear</u>, <u>joining the FSF</u> as an associate member, or making a **donation**, either <u>directly to the FSF</u> or <u>via Flattr</u>.

back to top

Please send general FSF & GNU inquiries to $\leq gnu@gnu.org \geq$. There are also <u>other ways to contact</u> the FSF. Broken links and other corrections or suggestions can be sent to $\leq licensing@gnu.org \geq$. Please see the <u>Translations README</u> for information on coordinating and submitting translations of this article.

Copyright © 2014, 2015 Free Software Foundation, Inc.
This page is licensed under a <u>Creative Commons Attribution-NoDerivs 3.0 United States License</u>.
Copyright Infringement Notification

Updated: \$Date: 2015/04/17 15:55:35 \$

The Software

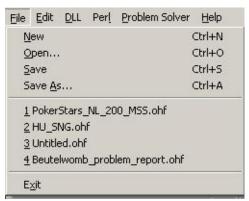
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:16:33.093000

The GUI

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:34:03.031000

Main Menu

File



The File menu is pretty self-explaining. It contains some standard Windows menu items to create a new document and to open and save the current formula set. Below the sub-menu shows a list of last recently used formulas which you can select to open. The last menu item finally will quit OpenHoldem

Edit



The edit-menu contains the following items:

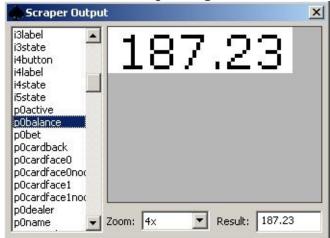
- *Formula*: this opens the formula-editor which will be explained in a separate chapter
- *Preferences*: this opens a dialog with option settings. Again it is so manifold that it deserves an extra chapter
- *Tag Log-File*: This will create an entry in the log that makes it easy to find and review critical situations.

[*** ATTENTION ***] User tagged this situation for review

If the autoplayer already acted this log-message will appear slightly after

the interesting decision.

- *View Log-File*: this will open the log-file of this OpenHoldem instance. A convenient feature if you have multiple logs in your directory.
- *View Scraper Output*: this menu item opens a dialog where you can inspect the result of all scraped regions



• *Shoot Replay-Frame*: this final menu-item will shoot a replay-frame, be it for the creation of a tablemap or for reviewing the decisions of your botlogic. Again replay-frames and the OHreplay application are explained in an extra chapter.

Problem Solver



The Problem-Solver is a tool meant to help newbies to get OpenHoldem running. It will execute certain checks and will output diagnostic messages and helpful hints. As newbies regularly need help and don't know where to get help this feature got a prominent place in the main menu without being hidden in a

sub-menu.

Help



Finally the Help menu contains five menu items:

- *Help*: this option will display OpenHoldems main manual, i.e. this document.
- *OpenPPL Manual*: opens the OpenPPL manual, which historically became and still is a separate document. Most people writing bot-logic probably don't need to see the full complexity of OpenHoldem.
- *Forums*: this will open your favourite web-browser and lead you to the OpenHoldem forums at www.maxinmontreal.com/forums/index.php
- *Problem Solver*: this menu item will start the Problem Solver already discussed above
- *About this Program:*



Toolbars

OpenHoldem Main Toolbar



From left to right, the buttons on the Main toolbar are:

- New formula file
- Open formula file
- Save formula file
- Engage (or stop) the Autoplayer (nested diamonds)
- Edit current formula
- Enable validator. The validator is a tool that analyzes built-in symbols to detect inconsistent game-states that are mostly caused by bad tablemaps or bad settings. It helps you to test your tablemap before you take it to the realmoney tables.



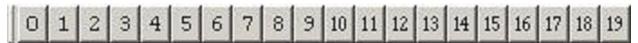
• Tag log-file. This will create an entry in the log that makes it easy to find and review critical situations.

[*** ATTENTION ***] User tagged this situation for review

If the autoplayer already acted this log-message will appear slightly after the interesting decision.

- Show scraper output
- Shoot a replay-frame
- Start ManualMode for offline simulations and testing
- Show the help-file (this file!)

OpenHoldem Flags Toolbar



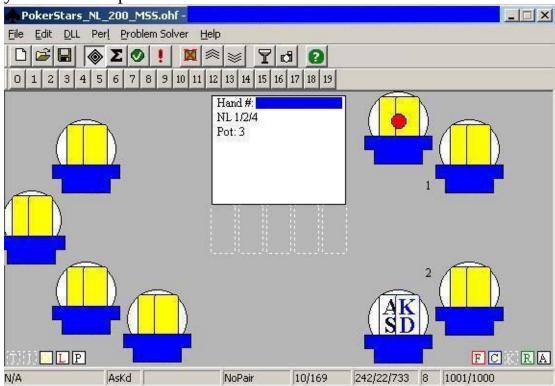
The Flags toolbar is used to set internal OpenHoldem flag symbols f0..f19 to on or off. These flags can be used in your formula set to quickly enamble or disable certain functionality. E.g. you could use flag 0 to to tell your bot to use a different starting-hands-chart or to do more conti-betting.

##f\$preflop##
WHEN f0 RETURN f\$PokerTragedySHCAction FORCE
WHEN Others RETURN f\$SklanskySHCAction FORCE

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-23T20:49:08.046000

Table Display

The OpenHoldem table display is intended to give you a quick visual reference as to what the screen scraper engine is actually seeing on the poker table at any given time. In this screen shot, OpenHoldem is connected to a PokerStars table. But as a beginner you might prefer the ManualMode application to familiarize youself with OpenHoldem.



The center information box contains various other information, including the hand number, limit type and limits, ante and current pot value. The table display will show all scraped chairs, i.e as many as specified by the nchairs-symbol in your tablemap. Chair 0 will be displayed at top-right, i.e 1-clock-position. For each chair the following information will be displayed:

- name
- balance
- bet
- cards or cardbacks
- dealerbutton
- sitting in "active" (white circle background) or sitting out "inactive" (grey circle background). Be careful! If you don't get these regions right in your

tablemap the scraper will be smart and optimize some of the more costly and now "superfluous" operations away. Luckily enough recognizing players sitting in or out is usually an easy and reliable colour-transform.

The FCKRA boxes on the bottom right indicate the status of the Fold, Call, Check, Raise and Allin buttons; colored means those buttons are seen as ready to be clicked, and white dotted outlines means they are not seen as ready to be clicked.

Very similar, but not used by every tablemap, are the TIOLP buttons at the bottom left. They display if you can click autopost, sit in, sit out, leave.and prefold.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-30T00:33:25.531000

Status Bar

At the very bottom of the OpenHoldem GUI you will find the statusbar.

Thinking Kc5c Qd5h7s OnePair 55/169 134/14/849 8 1000/1000

From left to right, the fields on the Status Bar are:

- Action taken
- Your hole cards
- The board cards
- Your made poker hand
- Handrank of your hole-cards at a level from 1 to 169 where 1 (AA) is best and 169 is worst against the number of opponents you have
- PrWin, PrTie and PrLos (the values mean: winning-percentages * 10)
- Current value of f\$number_of_opponents_for_prwin, one of the standard formula functions. This formula is intended to specify the number of opponents used for the prwin-calculations; however this number can be more or less than the number of opponents actually seen on the table at any given time. More information on f\$number_of_opponents_for_prwin can be found in the section about Primary Functions and in the section about PrWin.
- Number of iterations done and number of iterations to be done

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T03:56:06.046000

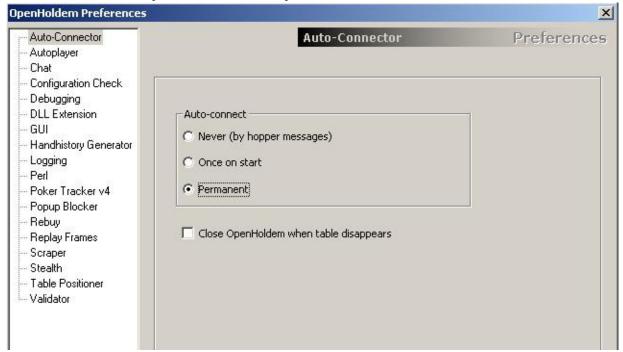
Preferences

The preferences for OpenHoldem's operation can be accessed via the Edit/Preferences menu item from the main window. A description of each preference follows.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:37:24.296000

OpenHoldem Preferences – **<u>J</u>Autoconnector**

Autoconnector preferences control how the OpenHoldem Autoconnector functions. See the {The Automation} section for more information.



• Auto-Connect

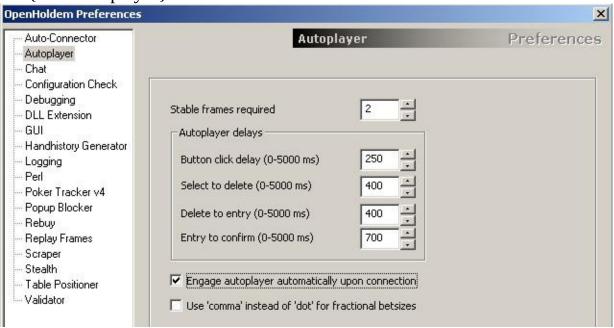
- Never: The former default but now outdated classical method. The clumsy and error-prone manual-connection is gone, but this option is still supported for the rare case that anybody wants to control all connections with his hopper. Please refer to the chapter "Automatization" for more details.
- Once on start: This option will cause OpenHoldem to auto-connect exactly once on start. This setting is especially useful in combination with the option to automatically choose a table. If no table exists, that fits to a tablemap or if all such tables are already getting served , it will fail silently and behave the default way, waiting for a connection-message.
- Permanent: This setting will cause OpenHoldem to try to auto-connect once every heartbeat. After some refactoring OpenHoldem does now cache the connection-data of all known table maps and all instances of OpenHoldem are now synchronized. If one instance fails to connect, the other ones will be blocked for the next N seconds. So the system-

- load is now astonishing low, even with dozens of instances waiting for tables. Therefore this option became the new default.
- Close OH when table disappears: This option will terminate OpenHoldem, when the poker table stops to exist, e.g. when the table got closed by the hopper or when you get moved to another table in a MTT. This method is for you, if your philosophy to do reconnects is to close OpenHoldem and to simply fire up another instance or if you want to easily shut your botting system down at the end of a session.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:28:53.812000

OpenHoldem Preferences – ↓Autoplayer

Autoplayer preferences control how the OpenHoldem Autoplayer functions. See the {The Autoplayer} section for more information.



- Stable frames required: This parameter specifies how many additional simultaneous identical screen scrapes must occur prior to the Autoplayer taking action. Generally, you want to have a stable poker screen before your take action, and this setting will allow various animations and so forth to finish, ensuring that your poker logic has correct game state information prior to making a decision. A frame-delay of 0 would mean that OpenHoldem acts immediatelly after a scrape contains your cards and action buttons. Each additional frame will require a "full" heartbeat, but as soon as it is your turn the heartbeat-pause will be reduced by 50% to achieve a better combination of stable frames and fast response time.
- *Autoplayer delays:* These delays control the little pauses inside a more complex autoplayer action, for example the time between entering a betsize and clicking the raise button.
 - *Button click delay:* This setting contriols the delay (in milliseconds) between multpile button clicks (double and triple clicks).
 - Select to delete: For betsize actions, this setting determines how long (in milliseconds) that OpenHoldem will pause after performing the betsize select action as specified in the Table Map, and prior to

- performing the betsize delete action as specified in the Table Map. These preferences are not set in the Table Map itself, as these are highly dependent on your CPU and network capacity and utilization, and thus are largely specific to your particular computing environment.
- Delete to entry: For betsize actions, this setting determines how long (in milliseconds) that OpenHoldem will pause after performing the betsize delete action as specified in the Table Map and prior to performing the betsize entry action. These preferences are not set in the Table Map itself, as these are highly dependent on your CPU and network capacity and utilization, and thus are largely specific to your particular computing environment.
- Entry to confirm: For betsize actions, this setting determines how long (in milliseconds) that OpenHoldem will pause after performing the betsize entry action, and prior to performing the betsize confirm action as specified in the Table Map. These preferences are not set in the Table Map itself, as these are highly dependent on your CPU and network capacity and utilization, and thus are largely specific to your particular computing environment.
- Engage Autoplayer automatically upon connection: This setting instructs OpenHoldem to automatically start the Autoplayer immediatelly after it connects to the table. It is especially no longer necessary to identify the userchair for that. This way it becomes possible to control the option to auto-post blinds and do other minor tasks.
- *Use 'comma' instead of 'dot'*: Some casinos require the use of the "comma" as a decimal separator rather than the "dot". This setting will instruct OpenHoldem to use a comma if needed. Eventually this setting should be moved to the Table Map file itself, as it is casino-specific.

In the past OpenHoldem supported some more autoplayer options, namely acting only with known cards, acting only with known balance (potential occlusion of the table) and cancelling autoplayer-actions that got interrupted by other applications or other tables that pop up (so-called loss of input-focus). These options got removed because it makes sense to have these sanity-checks and stability-features always enabled.

OpenHoldem Preferences – **↓Chat**

Chat preferences control how the OpenHoldem's Chat feature in the Autoplayer functions. See the {f\$chat} section for more information. The chat feature will generate a message if the f\$chat-function returns non-zero and the safety-delays are passed.



- *Minimum delay*: This is the minimum amount of time that Chat will wait (in seconds) between the posting of messages to the chat box on the Casino client. This is a safety feature and is present to prevent accidental spamming of the chat box. Feel free to set this to "1" if you are overly confident in your configuration skills.
- *Random delay*: In addition to the "Minimum delay", the amount of time specified in this parameter (in seconds) will be used to randomly delay the entry of a chat message into the chat box. The setting here is the maximum amount of random delay.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:31:35.531000

Configuration Checks

OpenHoldem will check some parts of your system configuration automatically on start-up to avoid the most common and time-consuming mis-configurations. Some of them (like minimum colour-depth) will always be executed, other can be turned off, if you don't use certain features.



- *Check Keyboard Settings:* this option will, check for an english keyboard layout. Keyboard settings affect especially the decimal point in numbers and therefore the scraper-engine and the auto-player. If you continue with incorrect settings, OpenHoldem may or may not work as expected. If you are an experienced user with a working setup you may turn this warning off, but if you encounter problems (especially with betsizing) or are a beginner, you should better get your keyboard layout configured the correct way.
- Check Window Theme Settings: this option will check, if the windows theme is set to Windows Classic (Windows 2000 style), as that is the recommended theme and many TableMaps are build using this configuration. If you know exactly, what you do and are able to adapt your TableMaps to other configurations, then you may turn this setting off. However be aware, that any fancy settings might possibly break the scraper and will reduce your possibility to exchange maps with other users.
- Check Font Smoothing Settings: This option will check, if Font Smoothing

is turned off, as recommended. Font Smoothing does display names and numbers slightly more eye-freindly for a human user, but it will render scraping more difficult. If you know exactly what you do you may again turn this option off.

Other Configuration Checks

OpenHoldem does execute various other configuration checks, but part of them can't be turned off, as they are mandatory settings and the other ones will be triggered automatically.

Mandatory checks

• Mimimum colour depth of 24 bits.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-07-27T16:34:33.312000

OpenHoldem Preferences - Debugging

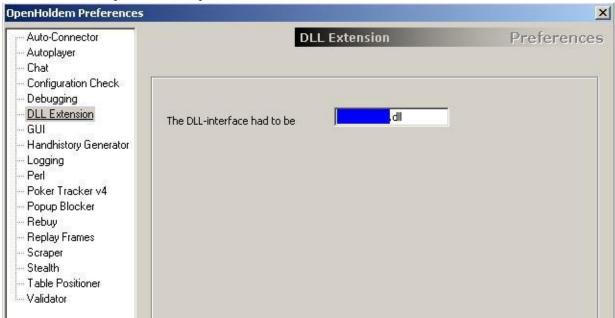
Debugging preferences control how much and what debug-messages OpenHoldem writes to the log-file. These options are especially meant for developers who sometimes need to get very detailed information about how OpenHoldem internally works when they try to fix a problem or test a new feature. Therefore we won't go into detail here. Just have a look and imagine what happens beyond the surface.

Auto-Connector Autoplayer Chat	De	Preferenc	
Configuration Check Configuration Check Debugging DLL Extension GUI Handhistory Generator Logging Perl Poker Tracker v4 Popup Blocker Rebuy Replay Frames Scraper Stealth Table Positioner Validator	Auto-connector Memory symbols Heartbeat-thread PrWin thread Blind locking Occlusion check Poker Tracker thread Rebuy management Replay frames Memory usage Session counter Stable frames counter Hand-history generator DLL extension Hopper-messages Popup blocker Symbol engine (OH-script) Symbol engine (OPenPPL)	□ board\$XZY expressions □ Autoplayer trace □ Autoplayer □ Tablemap loader □ Scraper directory monitor □ Table positioner □ Istournament □ AST priority ordering □ Table-limits (blind values) □ Lazy scraper □ Betsize adjustment □ Handreset-detector □ Engine container □ Preferences □ Scraper-preprocessor □ OpenHoldem.cpp □ Formula □ All the rest	Versus Scraper Filenames GUI ICM Tokenizer Parser

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:36:33.265000

OpenHoldem Preferences - JDLL Extension

DLL Extension preferences control how the OpenHoldem works with a User DLL. See the {User DLL} section for more information.

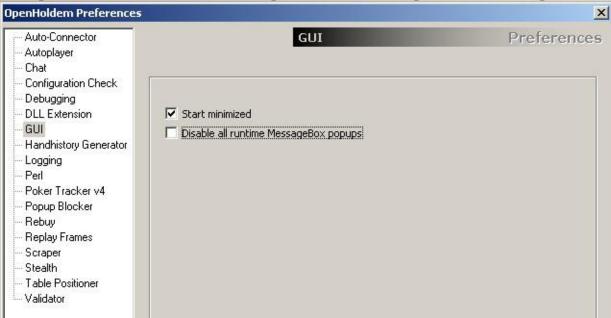


• *Name of DLL to load on startup*: This specifies a DLL file to load on startup. The DLL will always be loaded whenever this option is non-empty.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:37:38.187000

OpenHoldem Preferences – **JGUI**

GUI preferences control how the OpenHoldem GUI operates on startup.

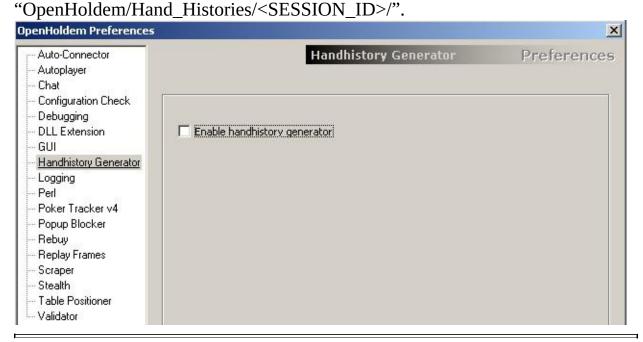


- *Start minimized:* This option will cause OpenHoldem to start in minimzed mode. It is especially useful, if you have to start new bots during a session and don't want to accept the risk of occluding tables and causing misscrapes.
- *Disable all runtime MessageBoxes*: This preference will prevent OpenHoldem from displaying warning and error message boxes during runtime (when OpenHoldem is connected to a table). This is extraordinarily dangerous and should only be used when you have a 100% stable bot that you want to move to a fully automated execution model. It is strongly advised that you do not check this option until you have a very stable and thoroughly tested botting environment. Interactive messages on user-interaction are not affected by this option.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:38:11.140000

OpenHoldem Preferences - Hand History Generator

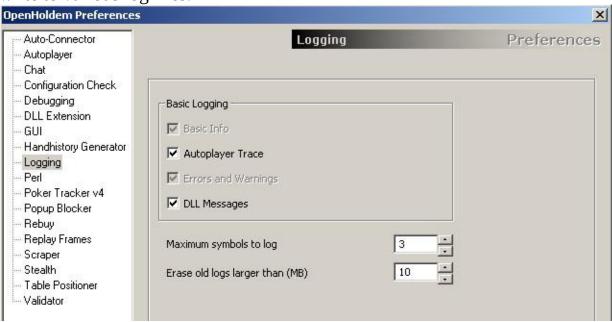
This preferences tab provides just a single option to enable or disable the handhistory generator. This option is meant to support poker sites that don't store hand histories at your hard drive for session review and PokerTracker support. If enabled the hand histories can be found in the directory



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:41:09.109000

OpenHoldem Preferences – Logging

Logging preferences control what information OpenHoldem will periodically write to various log files.



• *Basic Info*:This setting will cause OpenHoldem to log very basic information on every decision, giving you some hints what happened at the table and what OpenHoldem did. This information is given in a simple, beginner-friendly format, so that everybody can be able to understand what's going on. But for more detailed information you should enable the next option.

```
2014-09-30 07:11:37**** Basic Info ********************
 Version:
                 7.3.4
 Chairs:
                    10
 Userchair:
                     3
 Holecards:
Community:
Handrank:
                 5c5d
                 . . . . . . . . . .
  Handrank:
                 24
  Hand:
                 1pair
 My balance:
                   1000.00
 My currentbet:
                     0.00
  To call:
                     10.00
  Pot:
                     15.00
  Big blind:
                     10.00
  Big bet (FL):
                     20.00
  f$betsize:
                     30.00
```

Formulas: FC.RA Buttons: FC.RA

Best action: f\$betsize = 30.00

Action taken: f\$betsize

**** Autoplayer Trace *********************

• Autoplayer Traces: This option tells OpenHoldem to log the evaluation of formulas that lead to autoplayer actions, i.e the functions f\$alli, f\$betsize, f\$call for OH-script, the functions f\$preflop, f\$flop, f\$turn and f\$river for OpenPPL and also the evaluation of secondary functions like f\$sitout and f\$close. This information is very detailed and its interpretation will be discussed in the chapter about Logging.

- *Errors and Warnings*: This setting will write all errors and warnings to the log, which might be something like incomplete tablemaps,. syntax errors in your formula, run-time errors like division by zero or autoplayer errors like a missing input-box for the betsize. We consider this information so important that it will be always written to the log-file. No chance to turn it off.
- DLL Messages: Usually the autoplayer-trace will come to an end if a DLL-symbol gets evaluated as the DLL is a black-box for the OH-script-engine.
 However with this option enabled logging will continue even after DLL-symbols if the DLL calls back to OH-script to query the value of some OpenHoldem symbol or a user-defined function.
- *Maximum symbols to log*: This value defines how many special logsymbols shall be displayed in the white box in the GUI This affects userdefined log-messages that get printed when they get reached on evaluation:

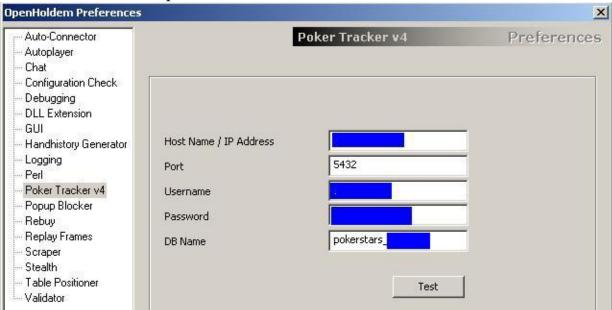
log\$WOOHOO_preflop_works

• Erase old logs larger than (MB): This option does specify a maximum size of existing log-files (in mega-bytes). OpenHoldem does usually append new log messages to existing files. Unfortunatelly these log-files can grow really large, especially if you did enable detailed logging. With this option old files which are too large will be automatically deleted, if OpenHoldem opens them on startup.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T04:01:40.453000

OpenHoldem Preferences - \(\precedet \) Poker Tracker

Poker Tracker preferences control how OpenHoldem interacts with a Poker Tracker database. See the {PokerTracker} section for more information on using Poker Tracker with OpenHoldem.



- Host Name / IP Address, Port, Username, Password, DB Name: These are
 the parameters that you entered when you setup your Poker Tracker

 ↓PostgreSQL database. Enter them here so OpenHoldem knows how to
 connect to your database. If you do not understand what these settings
 mean, please refer to the Poker Tracker homepage
 (http://www.pokertracker.com) for more information.
- *Test*: This button will test the connection to the Poker Tracker database as specified by the above parameters and will report success or failure codes. Help with connecting to PostgreSQL databases can be found here: http://www.pokertracker.com, or here: http://www.postgresql.org

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:43:44.656000

OpenHoldem Preferences - Popup Blocker

Popup Blocker preferences control if and how OpenHoldem handles any kind of popups. A popup is any kind of. "unexpected" window, i.e anything that is not

- the OpenHoldem GUI
- a MessageBox created by OpenHoldem
- a poker-table that fits to a tablemap in the scraper folder



The popup blocker has just three options:

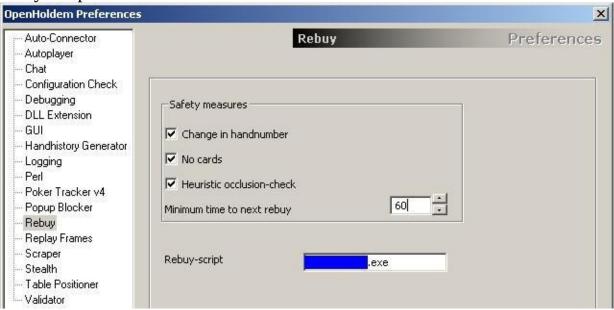
- *Diabled: no popup* will be closed. Choose this option if you want to handle popups with your hopper or if you are not afraid of any popups
- *Minimize popups*: any unexpecxted window will be minimized, but noit killed, Choose this option if popups are rare and you want to have a look at potential popups later.
- *Kill popups*: this will close any unexpected window immediatelly.

The popup-blocker is integrated into the auto-connector-thread, so you need one unoccupied instance of OpenHoldem that tries to connect to new windows if possible or to close everything that doesn't fit to a tablemap.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-02T21:53:42.250000

OpenHoldem Preferences - ↓**Rebuy**

Rebuy preferences control when and how OpenHoldem executes the external rebuy-script. See the section about Automation for more information.



- *Change in handnumber:* This option will cause OpenHoldem to wait at least for a new hand before doing a second rebuy. It is a simply sanity-check to prevent too many rebuys, as most casinos will increase your balance only after the end of the current hand.
- *No cards:* This option will prevent rebuys as long as you are involved in a hand.
- Heuristic occlusion check: Rebuys are especially volatile against misscrapes of the balance, which could be caused e.g. by temporary occlusion (The truths is: OpenHoldem will keep the last known balance, if it can't scrape the balance correctly. But it's always good to have another safety-net, isn't it?). The option to do heuristic occlusion checks will cause OpenHoldem to look at other players names and balances to guess, if heros balance is occluded. In case of a probable occlusion the rebuy-process won't be started. Heuristic occlusion checks will add some extra safety, but they won't catch every misscrape; e.g. the very unfortunate case of a partial occluded, but non-zero balance will still get accepted.
- *Minimum time to next rebuy:* This option is another sanity check to prevent to much rebuys in a short period of time. The time difference gets specified

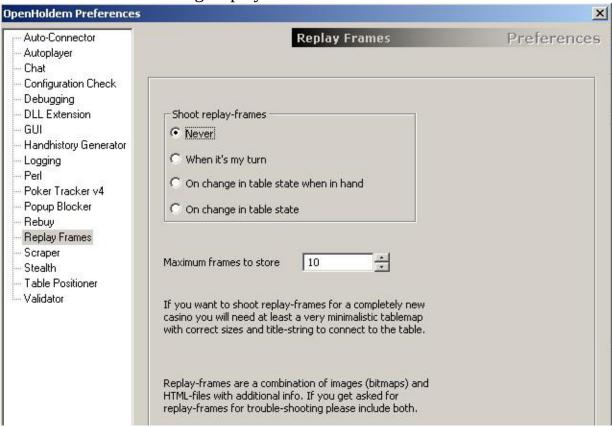
in seconds.

• *LRebuy-script:* This is the path to the external script that will be launched when a rebuy is required.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-31T07:37:28.312000

OpenHoldem Preferences - Replay Frames

Replay Frames preferences control how OpenHoldem collects Replay Frames for offline development and debugging use. See the {Replay Frames} section for more information on using Replay Frames.



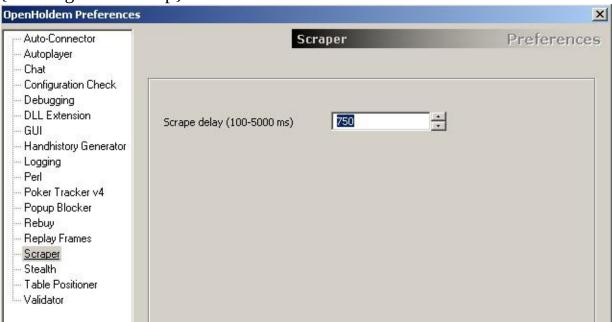
- *Never*: If this setting is checked, OpenHoldem will not record any Replay Frames.
- When it's my turn: If this setting is checked, OpenHoldem will collect exactly one replay-frame when it is your turn, exactly on the same heartbeat-cycle when the decision is taken. This is probably the most commonly used option, as it helps you to reproduce OpenHoldems decision making process.
- On change in tablestate when in hand: If this preference is checked, it instruct OpenHoldem to collect Replay Frames on every screen scrape cycle when you are in the hand. This option is usefuk to reproduce a complete game-history, but it avoids duplicate frames and thus saves space on your hard disk..

- On change in tablestate when in hand: Just lie the setting above this option will save every new replay-frame, but this time even if you are not in the hand. This might be usesful if you build your own stats-tracker, if you want to review your some of the secondary-formulas (f\$sitin, f\$close, ...) or if you want to build tablemaps that support these functions.
- *Maximum frames to store*: OpenHoldem will collect a maximum number of Replay Frames as specified by this parameter, then it will reuse filenames starting at the beginning.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-31T07:51:24.375000

OpenHoldem Preferences – **↓Scraper**

Scraper preferences control how the OpenHoldem Screen Scraper engine functions. At the moment there is only one preference available here. See the {Creating a Table Map} section for more information.

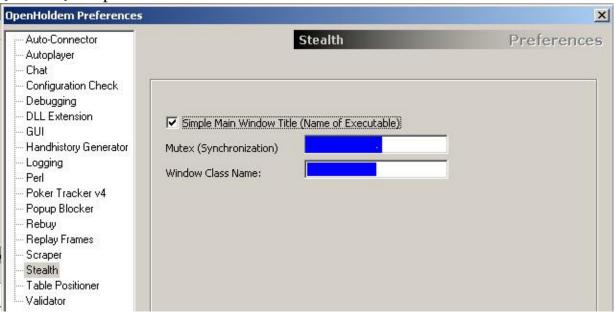


• *LScrape Delay*: This parameter specifies the amount of time (in milliseconds) that the OpenHoldem Screen Scraper engine will pause in between subsequent screen scrape passes. This amount of time does not specify the time from the beginning of scrape 1 to the beginning of scrape 2, but rather the amount of delay between the end of scrape 1 and the beginning of scrape 2. The primary consideration for this parameter is that of CPU consumption, and the default of 750ms seems to work well for most people.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-05T16:48:44.500000

OpenHoldem Preferences

Stealth preferences control various detectable components of OpenHoldem, and are primarily useful in setting up a highly stealthed environment. See the {Stealth} chapter for more information.



- *Simple Main Window Title*: By default, OpenHoldem will include all sorts of useful information in its window title bar. This useful information is also easily identifiable by the casinos' poker software. Check this preference to use a simple window title instead. OpenHoldem will then use the name of the executable in its title. For example if you rename the OpenHoldem binary to "ChessBase.exe" the title will be "ChessBase".
- *Mutex Name*: OpenHoldem uses system-wide mutexes to ensure that multiple running instances of OpenHoldem do not interfere with each other when performing Autoplayer actions. The name of this system-wide mutex is easily identifiable by the casinos' poker software and it is suggested that this be changed to something else.
 - *Remark:* You have to restart all instances of OpenHoldem when you chance the mutex name, otherwise the synchronization will fail.
- Window Class Name: Every window running under Microsoft Windows has
 a specific "class name" associated with it. This is easily identifiable by the
 casinos' poker software and it is suggested that this be changed to
 something else.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T04:02:39.671000

OpenHoldem Preferences - Table Positioner

Table Positioner preferences control if and how OpenHoldem arranges pokertables at the screen



- *Tiled*: this option will align all tables nicely without much overlap, so that OpenHoldem can usually see and scrape the tables clearly. Some overlap at the borders is possible however, and Microsofts TileWindow-function will also resize ann maximized windows.
- *Cascaded*: this option will stack the tables with lots of overlap, but usually in a way that a little piece of every table is visible. A useful option at the operating-systems that support scraping overlapping tables.
- *Never*: this option will disable the table positioner and leave it to the casino software or to your hopper to align the tables in a reasonable way
- *Always keep position*: this option will check once per heartbeat if the table is at its assigned position and move it back if needed.

If table-positzioning is enabled the lobby will get moved to the top left of the screen (0, 0) where absolute screen coordinates equal relative window coordinates. This makes your hopper more robust in case other windows pop up (lost focus). The lobby needs a minimalistic tablemap, just with clientsize(min/max) for connection, titletext and s\$islobby.

OpenHoldem Preferences – **↓Validator**

Validator preferences control various aspects of OpenHoldem's Validator engine. See the {Validator} section for more information. The purpose of the Validator is to help people to develop more reliable bots and Table Maps. It will execute several hundred ↓consistency-checks at the symbol level to detect invalid gamestates, ↓mis-scrapes or incomplete information. The Validator will write possible problems to the log-file and also show a message box, as long as messages are not disabled. The rules used by the Validator are always being refined and developed. The discussion forum is the best place to keep on top of this (http://www.maxinmontreal.com/forums/viewtopic.php?f=189).

OpenHoldem Preferences X Auto-Connector Validator Preferences Autoplayer Chat Configuration Check - Debugging Enable Validator - DLL Extension -- GUI Never Handhistory Generator When it's my turn -- Logging -- Perl C Always Poker Tracker v4 - Popup Blocker Rebuy Use heuristic rules Replay Frames Scraper Stop autoplayer on error - Stealth Shoot replay-frames on error - Table Positioner Validator The validator is a tool that tests the combination of * scraper * symbol engine * tablemap to detect incosistencies, usually caused by bad tablemaps.

- *Enable Validator*: This preference determines when the Validator is engaged to do its work, either never (completely off), when it is my turn, or always.
- *Use Heuristic Rules*: Some rules are heuristic, i.e. they are "common sense", but not always true. Example: A game does usually not last longer than 2 minutes, so the autoplayer has to act at least once per 120 seconds. If it does not do so, there seems to be a problem. Maybe the Table Map does

- not detect the players cards or the buttons. This setting can result in false positives and is recommended for testing only, but not for real-money-play.
- *Stop Autoplayer on error*: If the Validator discovers an inconsistency, and this preference is checked, the ↓Autoplayer will be instructed to stop autoplaying.
- Shoot replay-frame on error: If the Validator discovers an inconsistency, and this preference is checked, OpenHoldem will shoot a \preplay-frame and store it in the replay-folder. The replay frame will help you to find the source of the error.

It is never a bad idea to use the validator some time for testing new tablemaps. We recommend to enable it at your turn, because your turn is the most important time in a game: here the decisions will be taken, here perfect input matters most and here we usually have stable input, as the casino usually will no longer update the regions that need to be scraped. Give it a try, but be prepared — the validator is a little bitch!



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-23T20:50:45.968000

Formula Editor

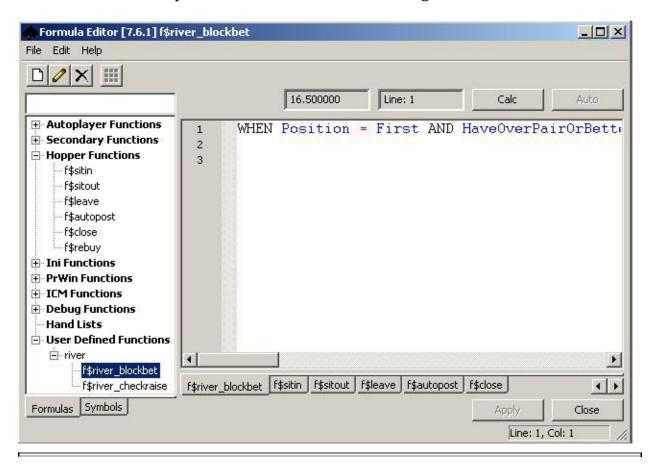
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-23T17:56:42.500000

Formula Editor

On the left you will find the treeview that is already populated with the deault OpenHoldem formulas, grouped into several categories.

The toolbar below the menu contains the following buttons from left to right:

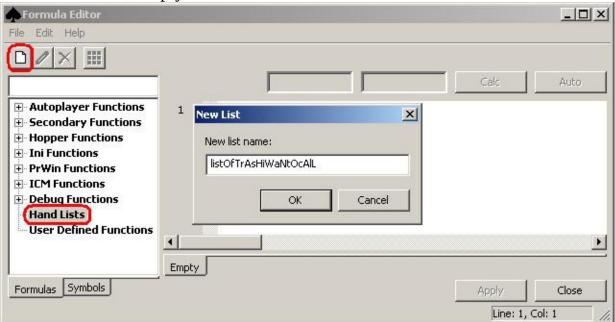
- *New*: creates a new user-defined formula or hand-list.
- *Rename*: renames an the selected user-defined formula or hand-list.
- *Delete*: removes an the selected user-defined formula or hand-list.
- *Edit handlist*: opens the hand-list editor to change the selected handlist



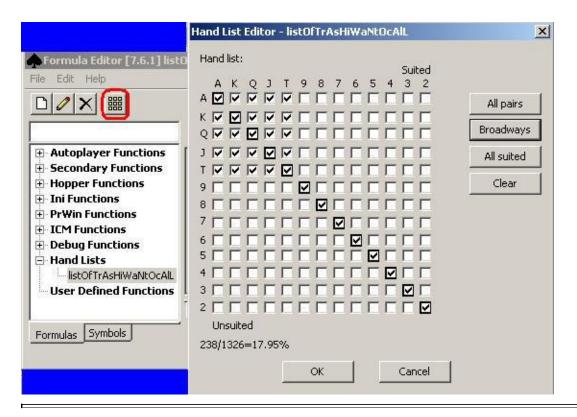
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T04:05:45.765000

Handlist Editor

OpenHoldem supports hand-lists, mainly for preflop decisions. Creating a new handlist works basically the same way like creating a new formula. Select the category "Hand Lists", then press the button with the blank sheet and a dialog will pop up that asks you for the name of your list. All names are allowed as long as they start with the prefix "list". Choose a name, press "OK" and the list has been created; empty for the moment.



To edit your newly created list select it in the tree-view and click the matrix-button in the toolbar. A dialog will pop up that allows you to choose the hands in the lists. Pairs are at the main diagonal, Suited hands above and offsuited hands below.



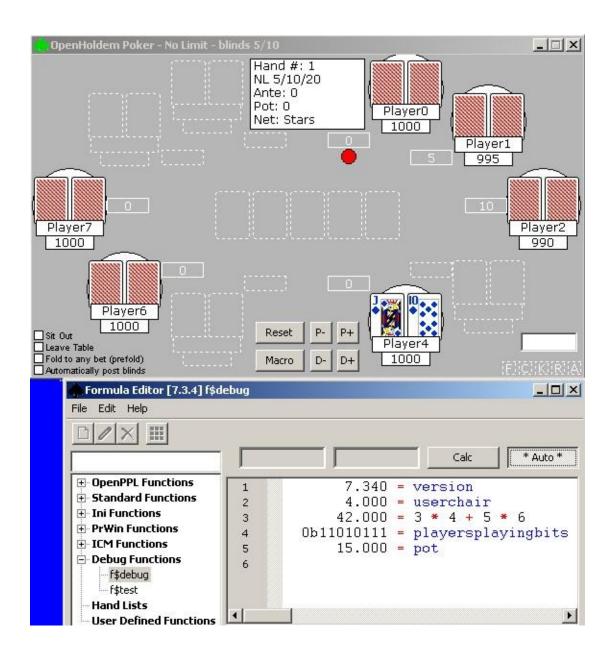
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-01T22:49:04.359000

Debug-Tab

A special "function" in the formula-editor is the debug-tab. It supports the evaluation of symbols, functions and mathematical expressions while being connected to a table, be it ManualMode, replay-frames or a life table. Using it is quite simple: one expression per line, each one after an equality-sign.

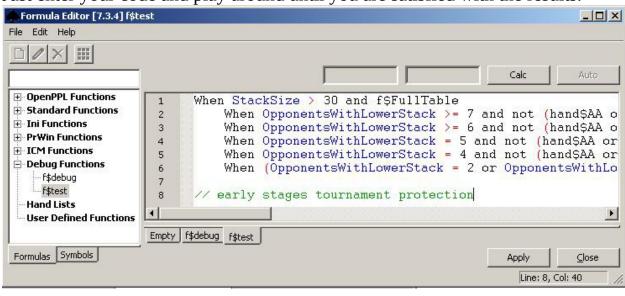


After you have entered the expressions you are interested in just click *Calc* or *Auto* and OpenHoldem will show you the results of all evaluations like in the picture below. The debug-tab is one of the most useful features for developing and debugging your OpenHoldem formula set.



f\$test

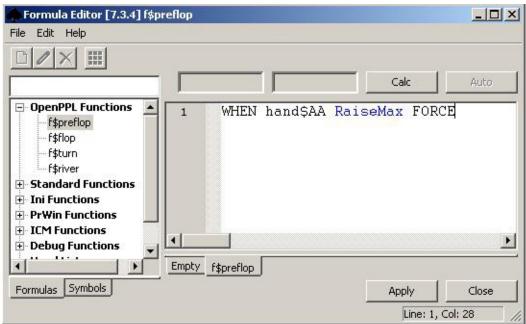
In addition to the debug-tab there is a special function named *f\$test*. While f\$debug is meant for a set of multiple expressions f\$test is for a single large expression, even up to complete multi-line function text. No equality-sign here. Just enter your code and play around until you are satisfied with the results.



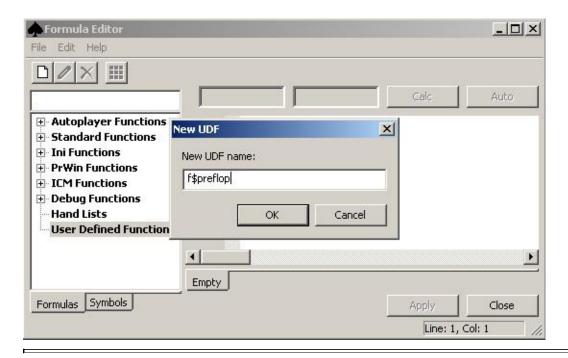
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-03T10:07:19.093000

OpenPPL-Mode

OpenHoldem uses per default the main autoplayer functions (f\$alli, f\$betsize, etc.). We call this default mode OH-script-mode. It is more or less compatibel to the former WinHoldem behaviour. However if you develop OpenPPL profiles the main autoplayer functions won't be used; OpenHoldem will use f\$preflop..f\$river instead and the formula editor will switch to OpenPPL mode as soon as the formula contains a *non-empty f\$preflop* function. Then all primary autoplayer-functions will be hidden and a subtree with the four main OpenPPL functions will be shown instead.



If your formula-editor is in OH-script mode then just create a new user-defined function, call it "f\$preflop", start editing it, apply your changes and reopen the editor.



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-01T22:53:38.890000

ManualMode

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:35:22.281000

↓ManualMode

Description

ManualMode is a tool for simulating a poker game state offine. It allows you to set any game state you can think of, including your cards, buttons, etc. A Table Map is included with the ManualMode tool to allow OpenHoldem to connect to the ManualMode main window and scrape the game state 100% correctly into the OpenHoldem engine. This will allow for offline construction of bot logic and the easy setup of various game state situations, some of which are very rare online (think Straight Flushes).

Familiarizing yourself with ManualMode

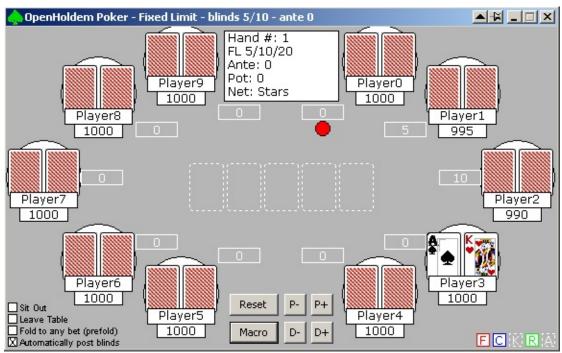


Figure 1.1 ManualMode

The ManualMode interface is quite intuitive. All relevant elements on the poker table are visible at all times, either populated with objects (like cards or bets), or outlined in white dashed lines if not populated. Each game state element is interacted with slightly differently and is discussed below. In general, however, right-clicking on any element will bring up a list

- 1. Common Cards: Right-clicking on each common card will bring up a context
- 2. Sit Down/Stand Up: OpenHoldem recognizes the difference between sitting in at a chair, and sitting in a game and being dealt a hand. allowing the player to Sit Down or Stand Up from the selected chair.
- 3. Sit In/Sit Out: OpenHoldem recognizes the difference between sitting in at a chair, and sitting in a game and being dealt a hand. cards will bring up a context menu allowing the player to Sit In to the game or Sit out from the game. table. up a context menu allowing the assignment of the dealer button to the selected chair. around the table anti-clockwise or clockwise.
- 4. Player Cards: Right-clicking on either of a player's cards will bring up a context menu allowing the selection of rank/suit, "cardbacks" or "no card". Note: prior to assigning cards, a player must have "Sat In" the game.

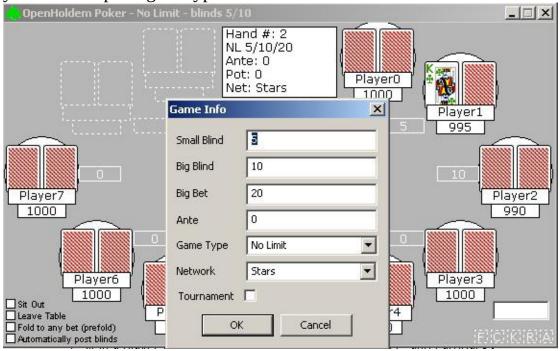
- 5. Player Names: Right-clicking on the player's name rectangle will pop up a window prompting for a string to use as the player's name. Note: prior to assigning names, a player must have "Sat Down" at the table.
- 6. Player Balances: Right-clicking on the player's balance rectangle will pop up a window prompting for a string to use as the player's balance. Note: prior to assigning balances, a player must have "Sat Down" at the table.
- 7. Player Bets: Right-clicking on the player's bet rectangle will pop up a window prompting for a string to use as the player's bet. Note: prior to assigning bets, a player must have "Sat In" the game.
- 8. Dealer Button: Right-clicking on either of a player's cards will bring up a context menu allowing the assignment of the dealer button to the selected chair. The "D-" and "D+" buttons can also be clicked to move the dealer button around the table anti-clockwise or clockwise.
- 9. Blinds/Limits: Right-clicking on the center information box will pop up a window in which the small blind, big blind, big bet, ante, game type (FL, PL, NL), network (useful if integrating with {PokerTracker}), and tournament indicator can be set. Buttons: Left-clicking on any game-state button on the table will toggle its state from on to off. These include the FCKRA buttons on the bottom right (fold, call, check, table, prefold, autopost).

There are some shortcuts built into ManualMode to allow easier setup of from the table. As players are added, they will Sit Down, Sit In, and will be assigned a name and balance. Players will be added from chair 0 to chair 9, depending on what is open, and removed in reverse order. for a string to be entered. The format of the macro string is described here at the bottom of this page. Betting: Left-clicking (right click to set a specific bet amount).

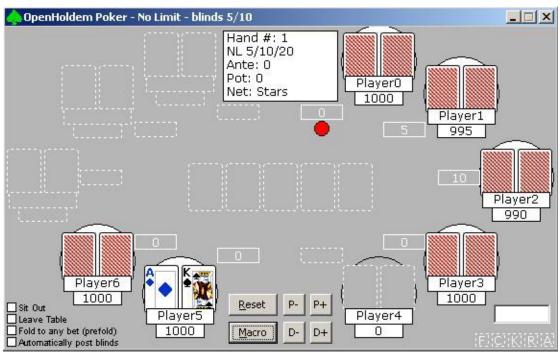
Note: When ManualMode detects a betting round change (i.e. 0 to 3, 3 to 4 or 4 to 5 common cards), then all player bets will be collected into the pot, and all player bets will be reset to zero.

Defining the game-type

If you right-clicking at the "white box" ManualMode will show a dialog where you can set up the gametype to be simulated.



Macros



A macro string is a series of characters executed one at a time, left to right. The meaning of each character is as follows:

Character Meaning

cdhs

R (uppercase) Reset the game state

N (Uppercase) Set the dealer button to this chair

P (uppercase) Sit in a player at this chair, default name, default balance,

and cardbacks

p (lowercase) Unseat a player at this chair

b (lowercase) Set the small blind at this chair

B (uppercase) Set the big blind at this chair

Set the rank of the players cards at this chair, or common

cards

23456789tjqka User cards have to be immediatelly after the player in

question,

all additional cards will be considered common cards.

See the suit of the players cards at this chair, or common

cards (player first, common next)

The default macro provided ("RNbBPpPAdKsP") used in the image above will

- reset (R) the table
- set the button (N) to chair 0
- set the small blind (b) to chair 1
- set the bigl blind (B) to chair 2
- set another player (P) to chair 3
- set an empty chair (p) to chair 4
- set another player (P) the user! to chair5 and deal him an ace of diamonds and a king of spades (AdKs)
- set another player (P) to chair 6
- finally leave the last 3 chairs will stay empty

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-07-27T11:32:13.750000

OHReplay

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:36:18.625000

↓Replay Frames

OpenHoldem has the ability to create "Replay Frames", either automatically as set in Preferences, or when the "camera" icon on the main toolbar {Toolbars} is clicked. A "Replay Frame" (or "frame" for short), is a combination of a Windows \(\pm\) bitmap-formatted image file (.bmp) and an associated \(\pm\)HTML file. These two files, in combination, provide an easy way to reproduce and diagnose a particular game situation. This is extremely helpful in the building and testing of your Table Maps and bot logic in an offline fashion. Simply connect OpenHoldem to a live poker table, and collect interesting frames for later use. Then at a later time, use \(\pm\)OHReplay to display these frames. Further connect OpenScrape or OpenHoldem to these OHReplay-displayed frames and work at your leisure.

The HTML provides crucial context for the frame, the most important of which is the title text. This text is used exhaustively in most Table Maps to understand the game state.

Often, when you post a problem or discussion on the forums, you will be asked to provide a frame and Table Map to help the developers diagnose your problem. Feel free to open the bitmap image in your favorite image editor (MS Paint?) and black out the sensitive portions if needed. Be aware that blacking out some parts of the image may prevent the developers or community from helping you, so be careful with that.

Both parts of the frame (.bmp and .html) and the associated Table Map are required to be provided when this is asked from you to help solve your problem! Frames are created in a "replay" directory directly under the directory from which you launched OpenHoldem. The .bmp files are large, so ensure that you have enough space to hold the maximum number of frames you specified in {Preferences}. Once the maximum number is reached, OpenHoldem will start reusing frame numbers from the beginning.

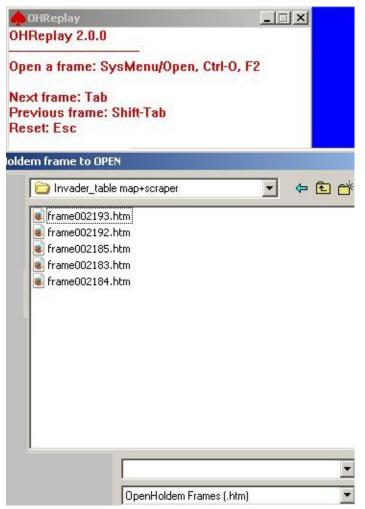
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-29T22:10:47.125000

↓OHReplay

Description

OHReplay is an application for displaying Replay Frames. The nice thing is that OHReplay displays frames exactly as they have been displayed by the casino. That means that once a frame is displayed by OHReplay, you can connect OpenHoldem to that displayed frame just as if you were connecting to the native poker client window. This allows for easy debugging of your Table Map and bot logic while offline.

Familiarizing yourself with OHReplay



The initial window displayed by OHReplay when it starts up contains all the information you need to know to navigate the program: To open a frame, use the System Menu, the Ctrl-O keystroke combination (that is a capital oh, the letter between N and P), or the F2 key. Any of these choices will open the standard Windows file open dialog, from where you can choose the \downarrow HTML file of the frame to open. To navigate between frames, use Tab or Shift-Tab to move to the next or previous numerical frame. To return to the initial screen, press Esc.

OHReplay Options

Clicking on the System Menu (red spades symbol) on the OHReplay window will bring up the system context menu. Selecting this entry will bring up the same file open dialog as using F2 or Ctrl-O.

Poker Tracker

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:36:47.828000

↓PokerTracker

Setup

Before you try to configure OpenHoldem to work with a Poker Tracker database, you must have a properly working Poker Tracker configuration set up first. This is not the most trivial of activities, and does have a learning curve. The best information on how to set up Poker Tracker can be found on the Poker Tracker website (http://www.pokertracker.com).

Assuming you have a working Poker Tracker installation, let's move on to getting OpenHoldem to work with your Poker Tracker database. OpenHoldem only works with ↓PostgreSQL databases, not with ↓Access databases; so if you do not have Poker Tracker configured in that way (and you know if you do), go back and get that set up first.

OK, so now you have Poker Tracker installed using PostgreSQL as the underlying database. Do you have Poker Tracker setup to automatically import hand histories into your database? If not, OpenHoldem will not be able to provide you any useful information, as there will be no information in your database to pull from. Learn how to automatically import hand histories here: http://www.pokertracker.com

After all of that is configured, you can use the settings in {Preferences} to instruct OpenHoldem how to connect to your Poker Tracker database. When connected, OpenHoldem will query the database using the \downarrow sitename and \downarrow playername as key parameters for the \downarrow query.

↓Sitename

Everything in Poker Tracker is first organized by what it calls a "site id". When querying the database, the first thing OpenHoldem will try to do is associate your sitename or ⊥network, as specified in the Table Map, to a Poker Tracker "⊥site id". This is done by doing a case-insensitive substring lookup against the following lists. This means that if your sitename string is "stars10sng" it will still be able to associate that to the proper "site ID", as "stars" is within that sitename.

If one of the below character strings can be found in either the sitename or network strings, as specified in your Table Map file, using a case-insensitive substring search, then OpenHoldem will be able to lookup statistics for the players playing at your table. As a note, the sitename string is searched first, and if a match cannot be found there, then the network string is searched.

- Stars
- Party
- FullTilt
- IPoker
- Everest
- Ongame
- Boss
- Cereus
- Pacific
- B2B
- MicroGaming
- Cake
- Bodog
- BetFair
- Cryptologic
- Ultimate
- Absolute
- WPEX
- Tribeca
- Merge
- Winamax
- EverLeaf

- YatahayENet
- Barriere
- Peoples

Playername

The second critical piece of information that OpenHoldem needs to query the database is the name of the player. OpenHoldem gets this from the pXname/uXname regions on the Table Map. Note that this scraped name does not need to exactly match the name as stored in the database. This is useful, as there is currently no method for scraping spaces in players' names. OpenHoldem will in all cases, return the statistics for an exact name match if it exists, and if it does not exist, will find the closest match using a Levenshtein distance algorithm (http://en.wikipedia.org/wiki/Levenshtein_distance). Note that Levenshtein distances that exceed a certain level (currently 30% of the length of the name) are discarded as invalid.

Table Maps

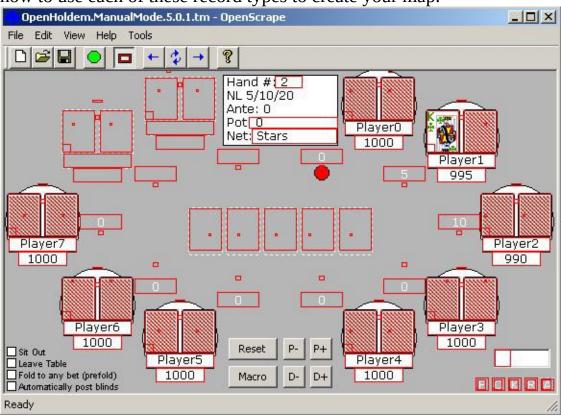
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:16:45.500000

Table Maps

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:38:47.968000

Creating a Table Map

Creating a Table Map is not a difficult process, once you get the hang of it, but it does have a steep learning curve. The benefit is that once you learn how to create maps, you can quite quickly take your bot to any casino on the Internet. OpenScrape is the program that is used to create Table Maps. It provides a graphical interface to provide a set of parameters that are stored in human-readable text files with the .tm extension. Try it — open any .tm file in Notepad and you can quite clearly see the way the file is laid out. Each Table Map has a set of parameters, grouped by record type. The following sections will discuss how to use each of these record types to create your map.



```
// OpenScrape 5.0.1
// 2014-07-04 16:36:34
// 32 bits per pixel
//
// sizes
```

.osdb2

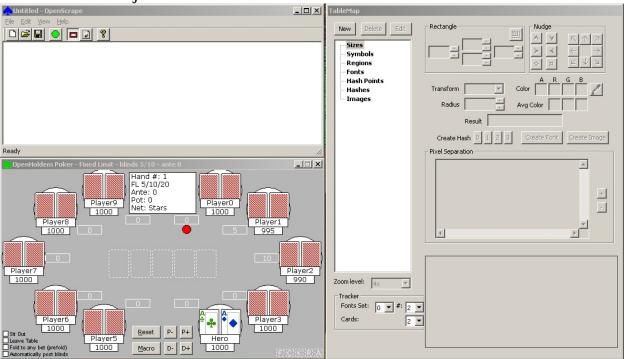
```
//
z$clientsize
                  574 335
//
// strings
//
                            10
s$nchairs
                            pokerstars
s$network
s$sitename
                            openholdem
s$betsizeconfirmationmethod
                               Click Bet
s$betsizedeletionmethod
                               Nothing
s$betsizeinterpretationmethod 3
s$t0type
                            fuzzy
s$titletext
                            Poker
                            ^*- ^L - blinds ^s/^b
s$ttlimits
                            ^*- ^L - blinds ^s/^b - ante ^a
s$ttlimits0
                            ^*- ^L tourney - blinds ^s/^b - ante ^a
s$ttlimits1
//
// regions
r$c0cardface0
                     193 144 224 185
                                                  0 H0
                                             0
r$c0cardface0nocard 213 170 213 170
                                        b6b6b6
                                                 10 C
r$c0cardface1
                     232 144 263 185
                                             0
                                                  0 H0
```

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-07-27T16:49:28.734000

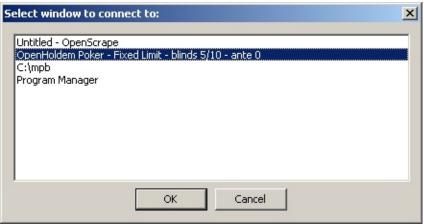
First things first: connect to the poker window

Start a casino, and open up a play money table. For the greatest part of building a table map you could even use "observer" mode if you like; only a few things like buttons will require a "real" account . Alternatively, once you learn how to use saved frames and OHReplay, you can also connect OpenScrape to these saved frames while offline from the casino.

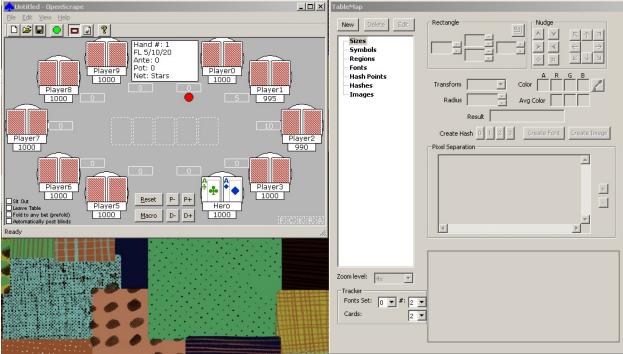
For now, let's keep it simple. Once you have a poker table setup, startup the OpenScrape program from the "c:\mpb" directory that you set up above. Arrange the casino table window and the OpenScrape windows on your screen in such a way that there is no overlap — OpenScrape needs a clear view of the poker table window to do its job.



Connect OpenScrape to the casino table by clicking on the Green Circle Button on the OpenScrape toolbar. A dialog with a list of top-level windows that are currently visible on your screen will appear. Select the one from the list that matches your casino table, and select OK. (Note, in this screenshot, we are connecting to a "ManualMode" window. You will learn more about ManualMode later in this document, but for now, just consider ManualMode to be a simulated, offline poker table).



A static image of the poker table window will be displayed in OpenScrape's main window. Once OpenScrape has the poker table's image displayed, you can minimize the poker table if you want, to give yourself more room to work. As the casino table changes state, and if you want to update the image that is displayed in the OpenScrape main window, you can do so at any time by clicking on the refresh button on the toolbar (Black and White arrows), or by choosing View/Refresh from the menu. When a refresh is requested, OpenScrape will bring the connected window to the front, grab the image, and then place it to the back again. Additionally, if you have OpenScrape connected to an OHReplay window, it automatically moves OHReplay to the next saved frame prior to grabbing the image.



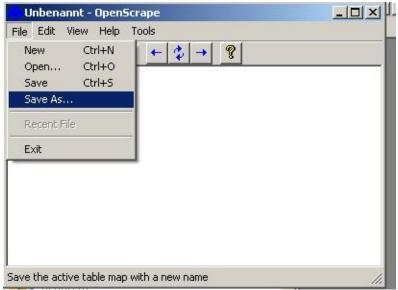
Once we have the poker table image in OpenScrape, it is time to start building our Table Map, and providing the parameters that the OpenHoldem game state

engine will use to "read" the poker table.

Familiarizing yourself with OpenScrape

When you started the OpenScrape program, two windows appeared. The "Main" window has the menu bar, the toolbar, and a display of the poker table image. The other window that appeared is the "TableMap Editor" window, and is where you will be doing most of your work with OpenScrape. The "TableMap Editor" window does not have a menu or toolbar, and has "TableMap" as its title.

File Menu



The File menu contains the standard Windows options to open and save Table Map files.

Edit Menu



The Edit menu contains the standard Windows options, plus two options specific to OpenScrape:

- Update Hashes: This option will recalculate all the Hash records in the current map useful if you have made changes to the underlying images, for example. More on Hashes later.
- Duplicate Region: This option will allow you to easily duplicate Regions. For example, once you have the player 1 card locations defined, you might want to copy those to players 2-10 to make things easier on yourself. More on Regions later.
- Clone Regions: This option will clone all regions of player 0 at once and even distribute them to reasonable positions at the table. All you have to do afterwards is fine-tune the positions a little bit. Prerequisite: the balance-regions of the other players must be defined in order to be able to determine the positions for the cloned regions.

View Menu



The View menu has following OpenScrape options:

- Connect To Window: This option duplicates the "Green Circle Button" on the toolbar, and will bring up a dialog to allow you to select which window OpenScrape will copy an image from.
- Previous Frame: if connected to OHReplay this option will switch to the previous replay-frame
- Refresh: This option duplicates the Black and White Arrow button on the toolbar, and will copy to current casino table image to the OpenScrape main window.
- Next Frame: if connected to OHReplay this option will switch to the next replay-frame
- Group Regions/Ungroup Regions: These options allow for customization of how Region records are displayed in the Table Map Editor window.
- Current size: Displays the client size of the window that OpenScrape is currently attached to. This is very useful when creating certain String records.

Main Toolbar



The toolbar on the main window performs the following actions, from left to right:

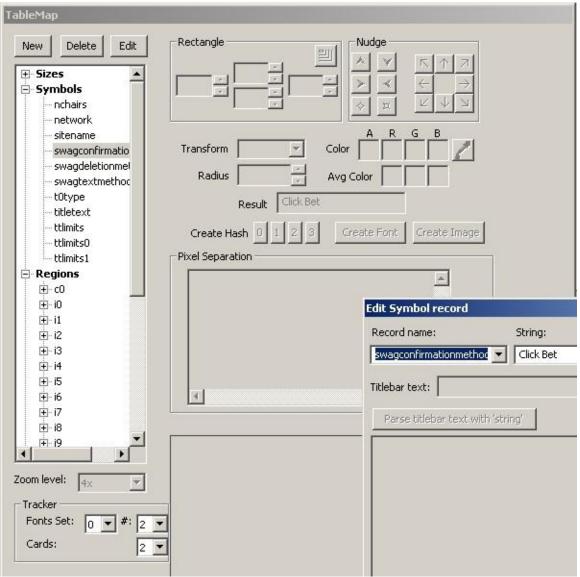
- New Table Map
- Open Table Map

- Save Table Map
- Connect to window
- Previous frame if connected to OHReplay
- Refresh the poker table image
- Next frame if connected to OHReplay
- Display Help/About

Keyboard shortcuts exist for many of these menu options and toolbar buttons, in addition to the standard Windows keyboard shortcuts:

- F1 Help/About
- F3 Connect to window
- F4 Toggle Region box display on or off
- F5 Refresh the poker table image
- Control + U Update Hashes

Table Map: Buttons/fields. The Table Map Editor window has a plethora of information on it, and various fields and controls will become enabled or disabled, depending on the type of map record selected.



On the top left of the window are three buttons, New/Delete/Edit, that allow you to add new records to the map, remove records from the map, or edit records in the map. The "New" button will prompt you to add a record of the type that is selected in the tree display directly below. For example, if you have a Region record selected, clicking New will prompt you to add a new region record to the map. The "Delete" and "Edit" buttons operate on the currently selected record. Directly below the New/Edit/Delete buttons is the record tree display. This box displays all records that are currently present in the map, and groups them by type. The Region records within this tree can be further sub-grouped using the View menu options on the main window. There are a number of choices for sub-grouping Region records, and are intended to improve the productivity of Table Map creation. Experimentation of these sub-groupings is necessary to see if any meet your personal needs.

Below the tree display is a drop down to select the zoom level of the image that is displayed in the big box on the bottom right portion of the window. The image displayed is dependent on the record type selected.

Below the Zoom Level are various drop downs that are used to serve as checklists for capturing fonts and card images. Capturing fonts and card images will be discussed later.

Creating and editing regions

On the top right of this window are a large group of controls, there is a "Rectangle" group, a "Nudge" group, and "Transform", "Radius", "Color", "Avg Color", and "Result" fields. These controls are used mainly for setting the parameters of Region records, and are only enabled when a Region record is selected in the record tree display on the left side of the window. The meaning of these fields will be described in the "Region" record section below. Some of these fields may display other relevant information, depending on the record type selected in the display tree. In addition to the arrow buttons a selected region can be moved and resized with hotkeys:

- arrow keys move it around
- arrow keys + shift moves it faster
- arrow keys + control makes it bigger/smaller (can also use shift for faster)

Below this large group of controls are buttons to create Hash records, Font records and Image records. These buttons will be enabled/disabled appropriately depending on the record type that is selected in the display tree. For example, creating an Image record only makes sense if a Region record is selected in the display tree. Creating a Font record only make sense if a Region record with a text transform type is selected in the display tree.

The box below the Hash, Font and Image buttons is used to display "pixel separation". The "+" and "-" buttons to the right of this box control the size of the font used in this box. Pixel separation is used for regions that use font transforms. A good separation between foreground and background colors is imperative for good font recognition, and this box lets you preview that separation.

Scraping in Practice

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:38:03.093000

Seated, active, playing

A player will be considered:

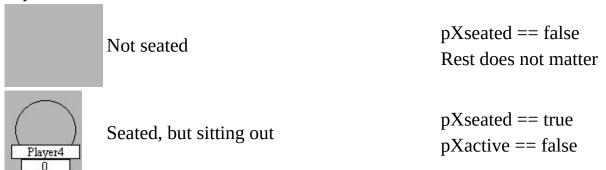
- *seated*, if his chair is occupied
- *active*: if he is posting blinds, i.e not sitting out
- *playing*, if he holds cards; i.e either face-cards or cardbacks.

OpenHoldem will scrape the above information in this order and much depends on getting this information right. For example if a player is not even seated, then OpenHoldem will be smart and avoid the costly scraping of cards and especially names. Fortunately deciding if a player is seated or active can usually be done with an inexpensive and reliable hash-transformation, which usually requires only the colour of a single pixel.

Scraping cards works in a similar way: It requires:

- cardbacks: again this information can usually be scraped with a cheap and reliable colour-transform
- "nocard"-regions. These regions were introduced for two reasons: if a player holds no cards we can avoid the somewhat costly scraping for cardfaces which usually requires fuzzy fonts. As an additional benefit we get improved error-detection in debug-mode, because a player either shows cardbacks, cardfaces or no-card.
- cardfaces (or alternatively ranks plus suits): this information can sometimes be done with hashes or images, if the cards are always displayed the same way: but often the display will differ a little bit and you will need text-scraping (fuzzy fonts).

In practice it looks like this:





Seated and active, but not playing

pXseated == true pXactive == true pXcardback == false pYcardfaceYnocard





Seated, active and playing (villain)

pXseated == true pXactive == true

pXcardback == true



Seated, active and playing (usually hero, at some casinos villain at showdown)

pXseated == true
pXactive == true
pXcardback == false

pXcardfaceYnocard

== false

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2017-02-25T19:25:05.140000

Starting a new map

Let's assume you start a new map completely from scratch. If you want to ease your job and get meaningful intermediate results we recommend to do your map in the following order.

- z\$clientsizemin/max and z\$targetsize first. They are absolutely necessary for connection.
- s\$titletext next. Juts like clientsize it is necessary for connection, because it might be that other applications or other casino tables have the same window size. Choose something unique like "No Limit Hold'em Logged in as". Once you have defined these two items and put this map into your scraper folder OpenHoldem will be able to connect to the table even if it does not yet scrape anything. but the autoplayer button should become active and you will get lots of warnings at load-time because of an incomplete map.
- s\$nchairs next. OpenHoldem needs to know how many chairs are supported by the tablemap. OpenScrapes region-cloner needs this info too if it clones the regions of player 0 to all other players.
- r\$pXseated for all chairs 0..n-1 or at least for chair 0 if you make use of the region-cloner thereafter. This is usually an easy hash returning true/false like the following three items.
- r\$pXactive to determine if a player is sitting in.
- r\$p0cardback, r\$pXcardface0nocard and r\$pXcardface1nocard. It is usually easy to do these hashes and you can now recognize all cards except your own hole-cards.
- r\$pXdealer to know who is the dealer.
- r\$c0cardfaceXnocard and r\$c0cardfaceX. The community cards
- r\$pXcardface0 and r\$pXcardface1. The player cards.
- r\$pXbet and pXbalance. This usually requires text-scraping and sometimes chip-scraping and is probably the most time-consuming part of the job.
- r\$iXbutton, r\$iXlabel and r\$iXstate. The basic action buttons
- r\$i3edit, s\$betsizeconfirmationmethod, s\$betsizedeletionmethod, s\$betsizeselectionmethod and s\$betsizeinterpretationmethod if you want to create a no-limit or pot-limit-map.

That's it. Now you are ready to start playing.

Getting started

This tutorial got written by a now unknown hero for our former Wiki (http://www.maxinmontreal.com/wiki/index.php5?title=Walkthrough). It describes a general roadmap for creating a tablemap from scratch given only openscrape and the target site/table window and nothing else. In this example we will pretend you are mapping a 10 chair Limit Holdem table at Pacific Poker.

- download openscrape (OH PC) and save it into a folder of its own , maybe "OpenScrape" .
- download Pacific poker (Casino PC) (http://setupspcp1.888.com/setups/support/full/en/3.509/pacificpokersetup.e create an account and open a Limit 10 seat Holdem Table.

The first step is to check the settings of the table, removing animations and other things that can be heavy or hinder the operation of OpenHoldem.



The next images shows some regions that will be used.

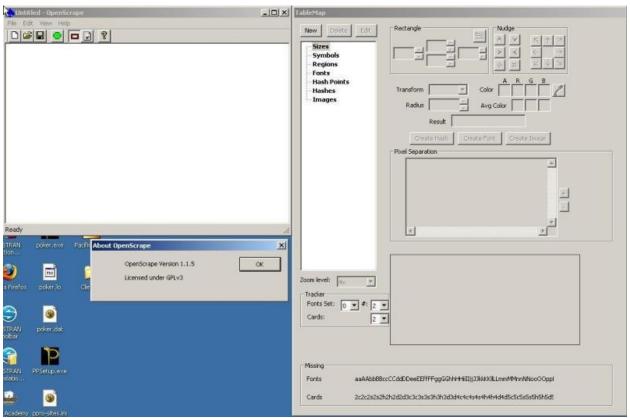


The player0 should always be the first to left of the dealer, which sits usually at the top of the table. The other players should be numbered clock-wise. Now open a bring client (OH PC) with the target site/table you want to map (Pacific 10 seat Limit). Bring will show you a table like the image below:



Once you have a bring-window open all operations can be executed savely at your OpenHoldem machine.

The next image shows OpenScrape running.



As a first step we will create a new TM and name it. Just click File --> New File --> Save As. Good naming conventions are:

- first the name of the casino in our example "Pacific"
- then the game type, as different game types do usually require different maps
- then the number of players
- then the version of the tablemap or the date of creation

In our example we will name the tablemap Pacific_NL_10_v0.1.tm. Now press the green button in OpenScrape and connect to the poker table.

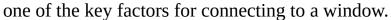


You should see openscrape changing to a copy of the poker window.



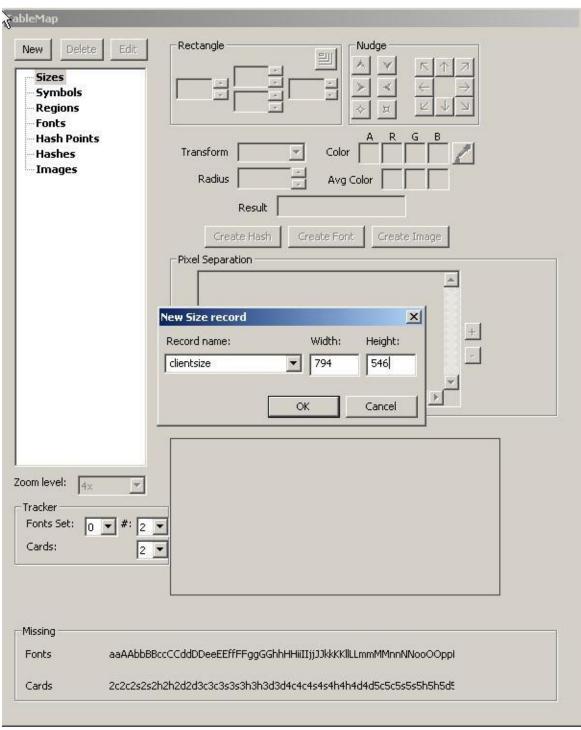
You can always use the refresh icon to shoot a new image of the poker window when you need to.

Now click the view menu and note the current size of the table. The table size is





Now expand the Sizes Tree in openscrape, highlight clientsize and hit Edit. Now add this information to your TM and press OK.

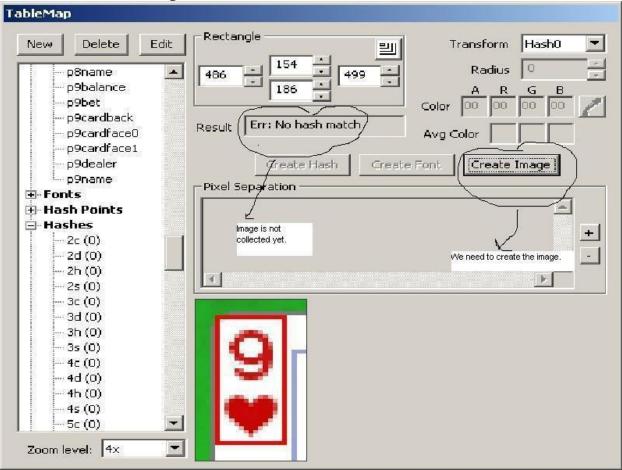


Remember to always save your TM after each modification. To create a complete map, we would now have to edit all symbols and regions. We will deal with this process in another tutorial in this manual.

Hashing Images

Using hash-values is the most easy way to recognize images. They are applicable whenever images get displayed in a fixed way. "Fuzzyness" like small changes in size and colour can't be handles with hashes. They require fuzzy fonts and cause more work, both for the creator of the map and at run-time. For this example we take a scraper-friendly casino.

First find the region box that contains the image that you wish to collect, and select the "Create Image" tab.

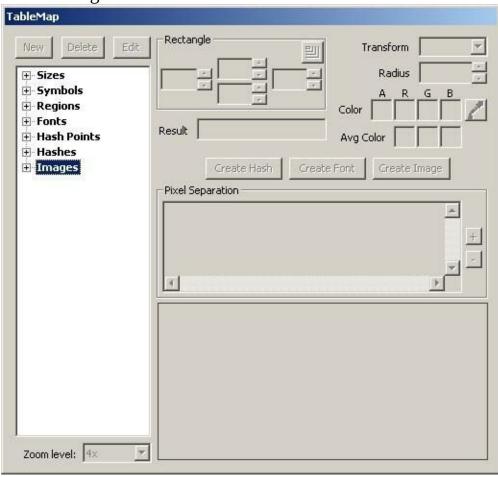


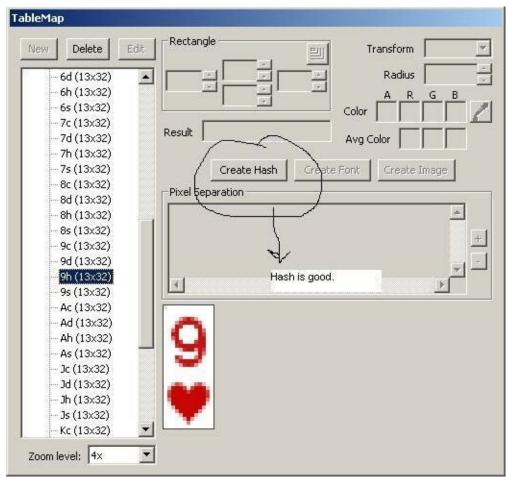
The pop up box for the name will appear. Give your image the proper name.



Go to the left side of Open Scrape and open the images tree and find your newly

created image and select "Create Hash".



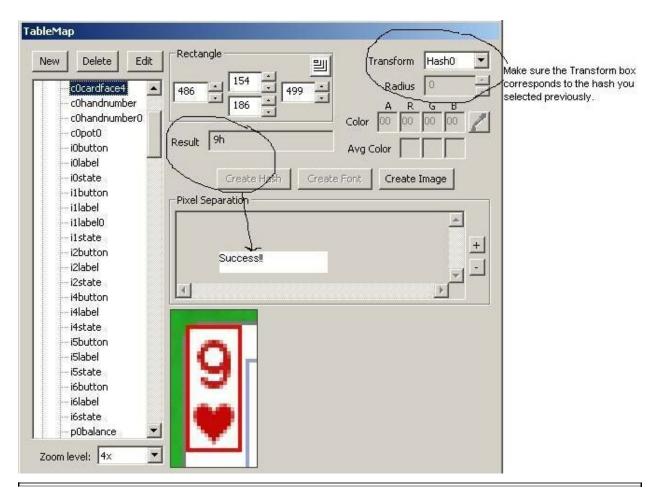


In most cases the hash value will be unique. Collisions can happen, but are extremeley rare. If that is the case you have to change your regions so that all images get an unique hash-value for identification.

Now select your hash type from the new pop up box. Generally starting with Hash0.



Go back to the Edit Menu on the Open Scrape Tool Bar and select "Update Hashes" from the drop down menu. Confirm the process was successfull by clicking on the box that contained the original image. As soon as you have selected the region OpenScrape should show you the desired result: the cards name, in our case nine of hearts.



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-19T09:31:09.484000

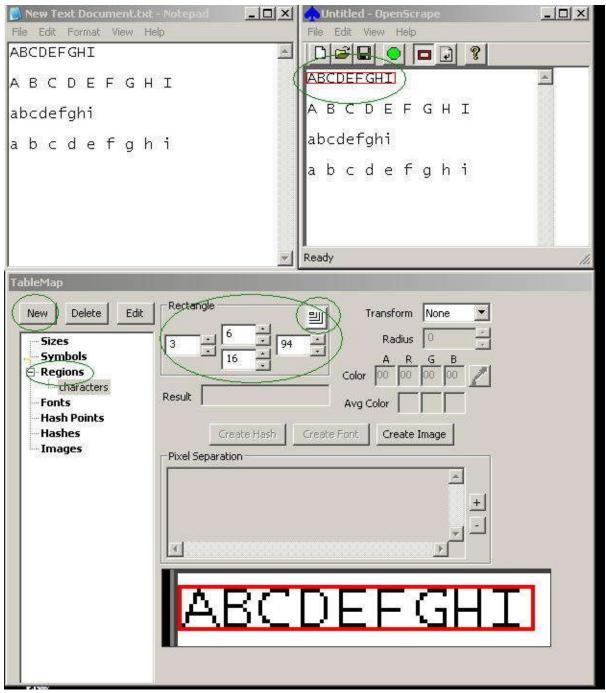
↓Collection fonts

This example on collecting fonts was reproduced from the Wiki (http://www.maxinmontreal.com/wiki/index.php5?title=Collecting_text).

We start with nothing except a table that we need to extract fonts from, here I am using a Notepad document for simplicity.



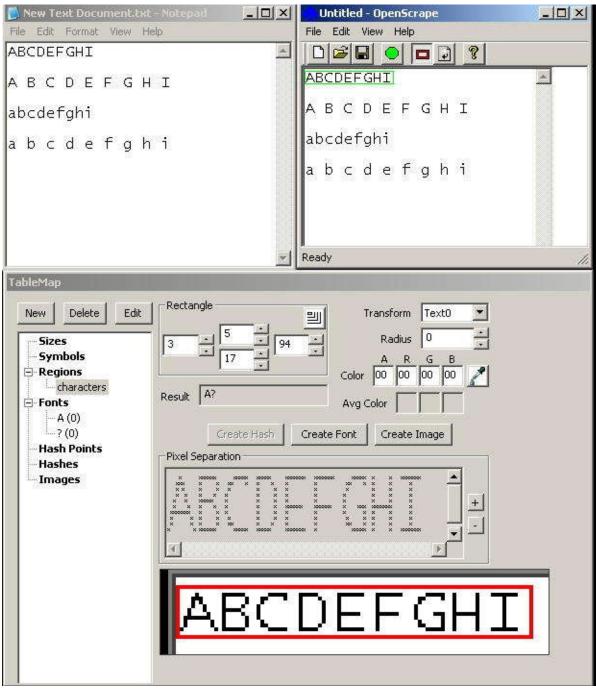
Next we fire up OpenScrape, and attach it to our "Table". Then we build a region around the area we are working on. Click "New" -> "Region" and name it characters. Use the coordinate editor to enclose the Region. You can also "shift-drag" the rectangle around to place it.



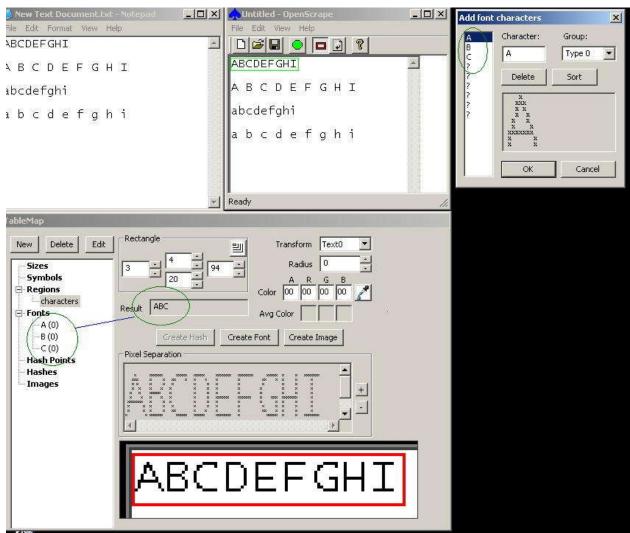
Set the region to the correct "transform", Type 0 in this case. Use the eyedropper to highlight the foreground pixels until we have nice seperation in the "Pixel seperation" field. Press "Create Font". We are presented with the "Add font characters" window. Highlight each "?" in turn and enter the correct character. "OK" to save. "Sort" when done here, then "save" on main GUI. If you hit the problem of openscrape seeing two letters as one because they have touching pixels then you will need to collect them seperately. Make a new temporary region that only encloses one character (call it char), capture the

Add font characters X ile Edit Format View Help Character: | D | = | | 0 | | = | | ? | BCDEFGHI Α Type 0 ABCDEFGHI BCDEFGHI Delete Sort ABCDEFGHI ıbcdefghi abcdefghi ıbcdefghi abcdefghi Cancel Ready ableMap Rectangle Delete Transform Text0 Edit Radius 0 Symbols ■ Regions Color 00 00 00 00 -- characters Result Err: No text match Fonts Avg Color **Hash Points** Hashes Create Font Create Image Images Pixel Separation

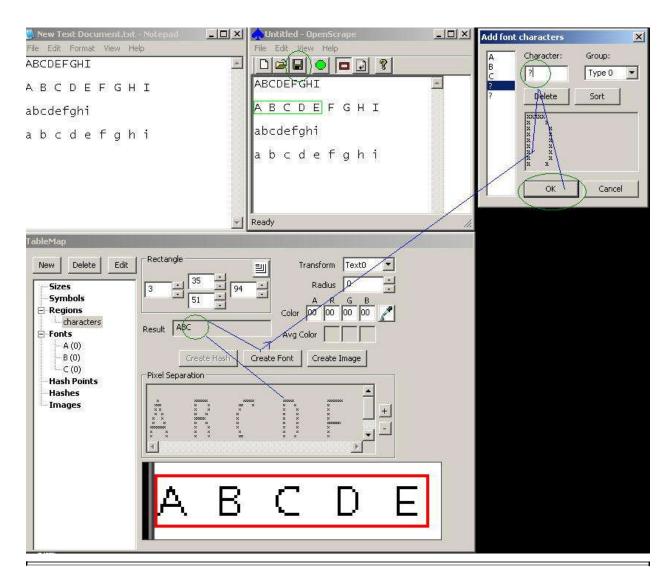
Here is the result after collecting only "A", you can see this by clicking inside the "characters" Region.



"A,B,C" are already present in the Table Map.



So on we go collecting fonts, here we collect "D".



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-19T09:32:08.515000

↓Scraping In Practice

How Do I Make A Map

This tutorial has been contributed by Ares® at http://www.maxinmontreal.com/forums/viewtopic.php?f=117&t=6547.

Here I will show you how I make a map from start to finish. We will concentrate on cards-scraping and the sometimes used chip-scraping here. The more easy text-scraping, which usually gets chosen for scraping the bets will be explained in another chapter of this manual. At first we check the version to avoid any surprises if the casino updates. These casino updates rarely affect the table-display, but sometimes they do and throw the scraper off.



Then we choose proper settings amd disable useless features like avatars and animations that could affect scraping. I also enable other features that ease scraping, especially the four-colour-deck.



Then we choose a nice table-view, the simpler the better.



This is how our poker table looks with our preferred settings



Checking this table display, we notice some good things and also some possible problems to care about:

- fonts are two colours, we have to use pXname / uXname for the names and pXbalance / uXbalance for the balance. In similar cases we might consider to use the background-colour in combination with a negative radius to accept both foreground-colours at once, but here we have two different background-colours, therefore this alternative approach can hardly be used.
- the dealer button is large and well positioned
- the cards are always exactly the same. Therefore we can use hashes, images or fuzzy fonts, but I will use hashes, as they are the most easy ones and the fastest.
- it might happen, that large chip-stacks occlude opponents card-backs.

Scraping Cards

There are three methods for collecting cards

- using hashes, i.e. using the image to compute a number and then doing a
 quick search for a match. This is the most simple method, but it requires
 that each card gets displayed exactly identically, at all times and at all
 possible places. We will ise hashes here, as Full-Tilt is a bot-friendly casino
 from a scraping point of view.
- using images. This approach is similar to the one above, but it requires more resources. This approach will be discussed in another chapter of this manual.
- using fuzzy fonts. This is "true screen-reading", somewhat more complicated, but it is possible to recognize cards that get displayed each time a little bit differently. As already mentioned in the introduction we will discuss this method in another chapter too.

So let's start with the card-scraping. We have five common-cards and two hole-cards in the situation below.



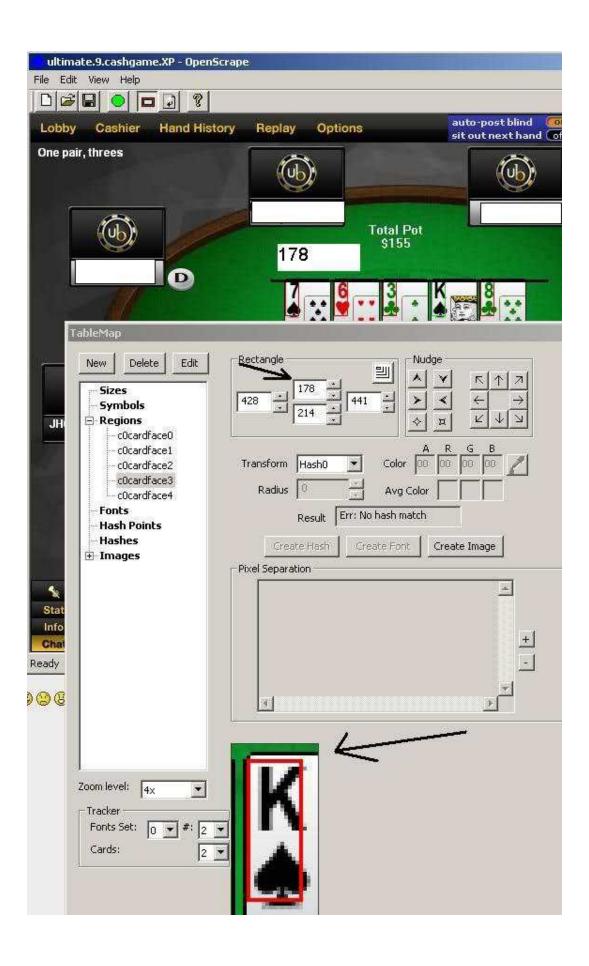
As you can see I pre-created five regions for the common cards, but in order to be able to define their size and position we have to care about my hole-cards. One of them is partially occluded and we would like to recognize all cards with the same hashes, therefore this card defines the size of our scraping region.

```
width 381 - 368 = 13
height 431 - 395 = 36
```

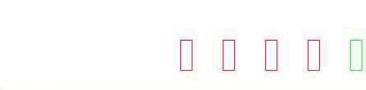
Please have a look at the region marked by arrow 3. The left border of the region matches the border of the card, but at the top is a free distance of one pixel. That's how we will align the regions for all other cards too.

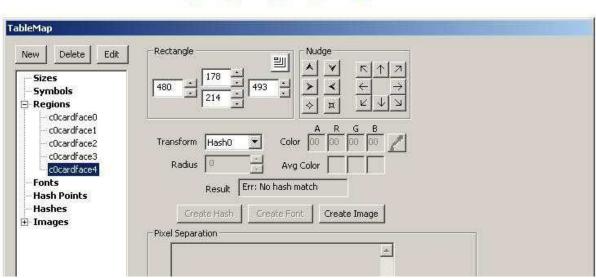


Next step: we take the size of the card-scraping region (13x36) and apply it to the common cards. All regions now appear with the correct size. but in the upper left corner (0, 0), as the position is still undefined (please see the blue arrow). Now we select these regions and click-drag them to the right places

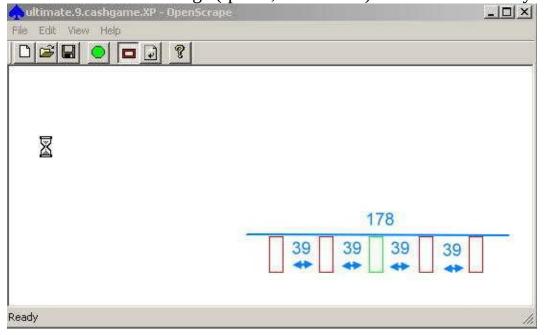


Now the common-cards are done. Always make sure that we use the same alignment for all cards. $\[\]$

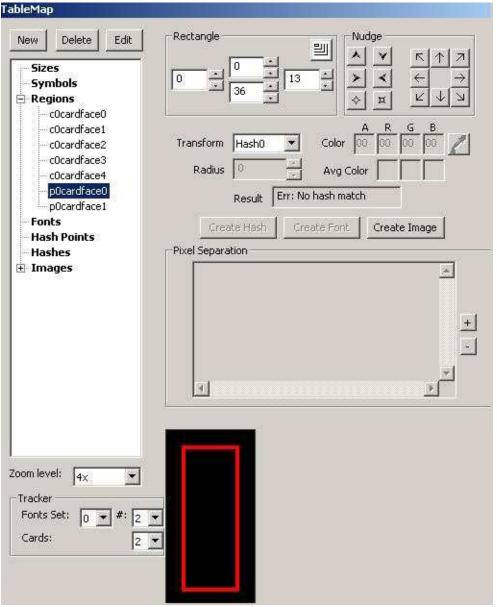




Now we check the alignment of the common cards. All regions should have the same height. The space between two regions should be equal. If the position of the fourth card can change (Ipoker, CakePoker) we should use fuzzy fonts.



After I have aligned the common cards I will create and align the cards for all players. Once this got done, I will seek a play money table, the most loose one I can get. This is one of the secrets of quick TM creation; I already managed to get 20 cards at once. So now I will create the regions p0cardface0 / p0cardface1. In OpenSrape I already created the region for the first player (player0). You can see that I chose the correct size of 13x36 and also chose hash0 transform. Now I save the tablemap and close OpenScrape to clone these regions with an external editor..



Here I opened the file in a text editor (notepad++ from <u>notepad-plus-plus.org</u>). You can see the two regions that we created for the first player.

```
14
    // strings
    11
15
16
17
    炒
18
    // regions
19
20
    11
2.1
    r$c0cardface0
                        272 178 285 214
                                                 0
                                                      O HO
22
23
    r$cOcardface1
                       324 178 337 214
                                                 0
                                                      о но
    r$c0cardface2
                        376 178 389 214
                                                 0
                                                      O HO
24
2.5
    r$c0cardface3
                        428 178 441 214
                                                 0
                                                      о но
26 r$cOcardface4
                        480 178 493 214
                                                 0
                                                      O HO
    r$p0cardface0
                                                 0
                          0
                              0
                                 13
                                      36
                                                      о но
    r$pOcardface1
                          0
                                                 0
28
                              0
                                 13
                                      36
                                                      O HO
29
    11
30
    // fonts
31
    11
32
33
34
```

Now I copy and paste the region's of the first player several times...

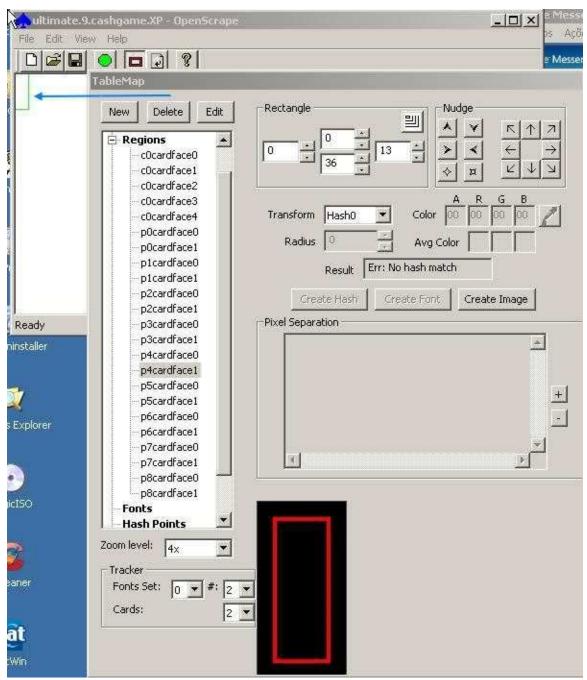
```
A 22
      r$c0cardface0
                        272 178 285 214
                                                    O HO
  23
      r$cOcardface1
                        324 178 337 214
                                                    O HO
      r$cOcardface2
                        376 178 389 214
                                               0
                                                    O HO
  24
      r$cOcardface3
                       428 178 441 214
                                               0
                                                    O HO
      r$cOcardface4
  26
                        480 178 493 214
                                               0
                                                    O HO
                                                    о но
  27
      r$p0cardface0
                          0
                              0
                                 13
                                     36
                                               0
  28 r$pOcardface1
                          0
                              0
                                 13
                                    36
                                               0
                                                    O HO
  29
      r$p0cardface0
                          0
                             0
                                 13 36
                                               0
                                                    O HO
  30 r$pOcardface1
                          0
                             0
                                13 36
                                               0
                                                    O HO
  31
      r$p0cardface0
                          0
                              0
                                 13
                                    36
                                               0
                                                    O HO
     r$pOcardface1
                                13 36
                                                    O HO
      r$p0cardface0
                          0
                              0 13 36
                                               0
                                                    O HO
      r$pOcardface1
                          0
                             0 13 36
  34
                                                    O HO
                          0
  35
      r$p0cardface0
                              0
                                13 36
                                               0
                                                    O HO
      r$pOcardface1
                          0
                              0 13 36
                                               0
  36
                                                    O HO
  37
      r$p0cardface0
                          0
                             0 13 36
                                               0
                                                    O HO
  38 r$pOcardface1
                          0
                             0 13 36
                                               0
                                                    O HO
      r$p0cardface0
                          0
                              0 13 36
                                               0
  39
                                                    O HO
  40
      r$pOcardface1
                          0
                              0 13 36
                                               0
                                                    O HO
                             0 13 36
  41 r$pOcardfaceO
                          0
                                               0
                                                    O HO
  42
      r$pOcardface1
                          0
                              0
                                 13
                                    36
                                               0
                                                    O HO
      r$p0cardface0
                                 13
                                     36
                                                    O HO
  43
      r$pOcardface1
                              0
                                 13
                                                    о но
                                     36
  45
```

...and rename the cloned ones, so that we get cardfaces for all 9 players (p0...p8).

Ø	2.0	777							
	21								
	22	r\$c0cardface0	272	178	285	214	0	0	HO
	23	r\$cOcardface1	324	178	337	214	0	0	но
	24	r\$cOcardface2	376	178	389	214	0	0	но
	25	r\$c0cardface3	428	178	441	214	0	0	HO
	26	r\$cOcardface4	480	178	493	214	0	0	HO
	27	r\$pOcardfaceO	0	0	13	36	0	0	но
	28	r\$pOcardface1	.0	0	13	36	0	0	но
	29	r\$p1cardface0	0	0	13	36	0	0	но
	30	r\$p1cardface1	0	0	13	36	0	0	но
	31	r\$p2cardface0	0	0	13	36	0	0	но
	3.2	r\$p2cardface1	0	0	13	36	0	0	но
	33	r\$p3cardface0	0	0	13	36	0	0	HO
	34	r\$p3cardface1	0	0	13	36	0	0	но
	3.5	r\$p4cardface0	0	0	13	36	0	0	но
	3.6	r\$p4cardface1	.0	0	13	36	0	0	но
	37	r\$p5cardface0	0	0	13	36	0	0	но
	38	r\$p5cardface1	0	0	13	36	0	0	HO
	39	r\$p6cardfaceO	0	0	13	36	0	0	но
	40	r\$p6cardface1	0	0	13	36	0	0	но
	41	r\$p7cardface0	0	0	13	36	0	0	HO
	42	r\$p7cardface1	0	0	13	36	0	0	HO
	43	r\$p8cardface0	0	0	13	36	0	0	но
	44	r\$p8cardface1	0	0	13	36	0	0	но
	45								
	150	60.60							

(On a side-note: cloning regions became more easy in the mean-time. Just create all regions for player0 and the balance-regions for all other players, plus the first common cards. Then click Menu \rightarrow Edit \rightarrow Clone Regions and all other regions will be created and even aligned to reasonable places. All you have to do is a little bit of fine-tuning the positions.)

In the next screenshot you see all cloned regions in OpenScrape, ready for positioning by drag and drop.



Be careful!



Today we received a gift from Ultimatebet. Their client has been updated, but it was only an update of internal functions, nothing that affected the user-interface. This is mostly the case with casino updates and one of the benefits of scraping: updates might happen often, but the interface changes rarely and mostly only slightly. But always be aware of updates.

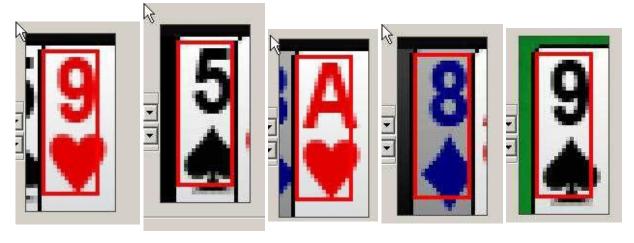
Now let's continue with the good work: drag and drop. Select the region you want to move, in the example "p7cardface0". The region will be flashing in yellow. Press the "Shift" key, put the mouse pointer over the region, click and move the region to the desired place. placing conventions: player0 sits at 1-o-clocl-position, the other players are seated clockwise..



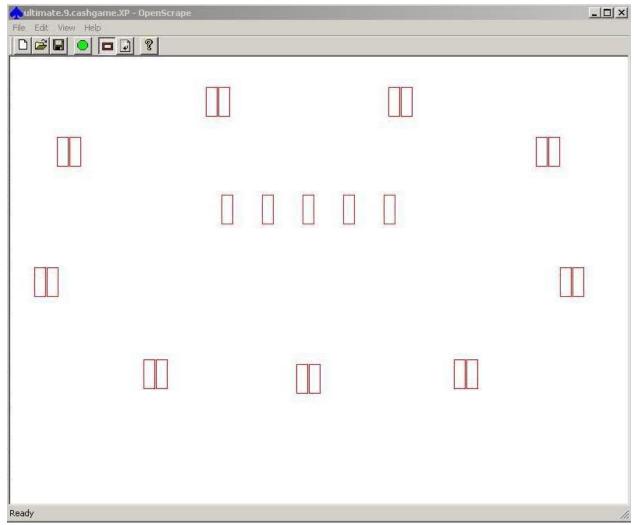
Do you notice the cards marked with arrows? They are not aligned properly. You can check this if you select them while they show cards.



We have to fine-tune these regions, so that all are aligned properly.



Checking alignment again. Pay close attention; if one region is misaligned, all data collected from this region will be wrong.



Now we start collecting cards

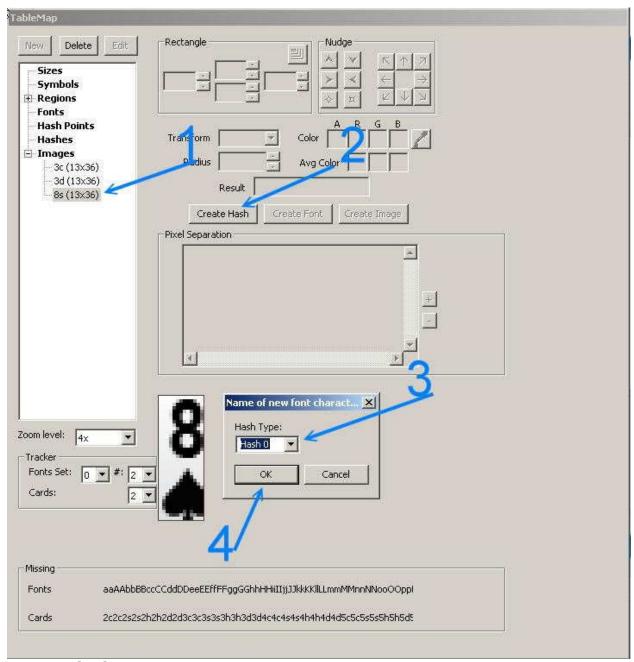
- 1. select the region
- 2. click "Create Image"
- 3. choose a good name like "8s"
- 4. click "OK"

Repeat this process until all cards are collected.



Creating hashes — step 1

- 1. select image (8s)
- 2. click "Creat Hash"
- 3. select "Hash 0"
- 4. click "OK"



Creating hashes — step 2

- 1. Click Menu → Edit
- 2. Click Update Hashes



Now OpenScrape already recognizes the card "8s"

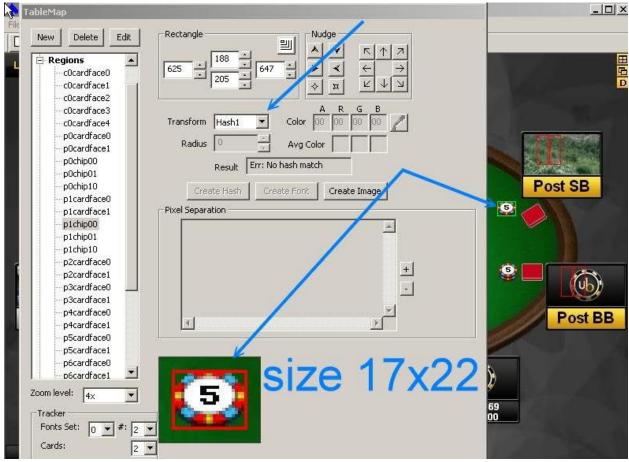


As the next step we would like to create the regions "pXbet", but the picture below shows an occlusion of "bets" by chip-stacks. Scraping the bet-amount is the preferred solution, as it is easy and doesn't utilize the CPU much. Unfortunately the occlusions prevents scraping the bet-amounts directly in our case. Therefore we have to use the alternative method "chip scraping", i.e. recognizing and counting chip-stacks. We will use the regions pXchip00, pXchip01 and pXchip10 for players, i.e. the bottom chip of the left-most stack, the 2nd-chip in the same stack and the bottom-chip in the 2nd stack. All other chip positions are then welldefined. The common pots work similar: c0potXchip00, c0potXchip01, c0potXchip10 for pot X.

,



Chip scraping



The timer of player8 covers the chip-stack partially. Therefore I will reduce the size of the scraped region to 22x7.



Then I open the TM in a text editor and pre-create the new regions, just like I did with the cards before.

41	r\$p7cardface0	61	104	74	140	0	0	но
	r\$p7cardface1	77	104	90	140	o	0	но
43	r\$p8cardface0	252	40	265	76	o	0	но
44	r\$p8cardface1	268	40	281	76	0	0	HO
45	r\$p0chip00	0	0	22	7	o	0	H1
46	r\$pOchipO1	0	0	22	7	o	0	H1
47	r\$p0chip10	0	0	22	7	o	0	H1
48	r\$p1chip00	0	0	22	7	o	0	H1
49	r\$p1chip01	0	0	22	7	o	0	H1
50	r\$p1chip10	0	0	22	7	o	0	H1
51	r\$p2chip00	0	0	22	7	o	0	H1
52	r\$p2chip01	0	0	22	7	0	0	H1
53	r\$p2chip10	0	0	22	7	o	0	H1
54	r\$p3chip00	0	0	22	7	o	0	H1
55	r\$p3chip01	0	0	22	7	o	0	H1
56	r\$p3chip10	0	0	22	7	0	0	H1
57	r\$p4chip00	0	0	22	7	0	0	H1
58	r\$p4chip01	0	0	22	7	0	0	H1
59	r\$p4chip10	0	0	22	7	0	0	H1
60	r\$p5chip00	0	0	22	7	0	0	H1
61	r\$p5chip01	0	0	22	7	0	0	H1
62	r\$p5chip10	0	0	22	7	0	0	H1
63	r\$p6chip00	0	0	22	7 7	0	0	H1
64	r\$p6chip01	0	0	22		0	0	H1
65	r\$p6chip10	0	0	22	7	0	0	H1
66	r\$p7chip00	0	0	22	7	0	0	H1
67	r\$p7chip01	0	0	22	7	0	0	H1
68	r\$p7chip10	0	0	22	7	0	0	H1
69	r\$p8chip00	0	0	22	7	0	0	H1
70	r\$p8chip01	0	0	22	7	0	0	H1
	r\$p8chip10	0	0	22	7	0	0	H1
77 TL	a bost place to a	1: 41	·	V -1-	:_VV	·	· •	41

The best place to align the pXchipXX regions is in the blinds, because there we get to see often single chips.

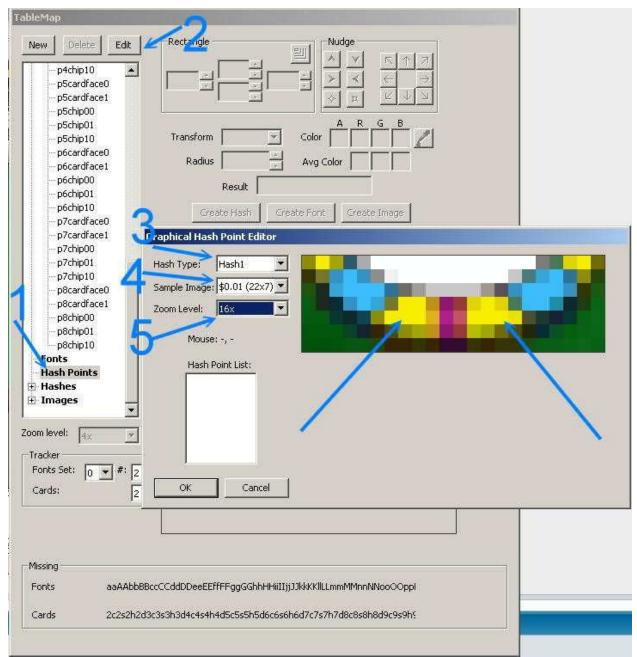


The chip should be collected as an image. The name of the picture must have a "\$" at the beginning (examples: \$0.01, \$0.25, \$1, \$5, \$100000)

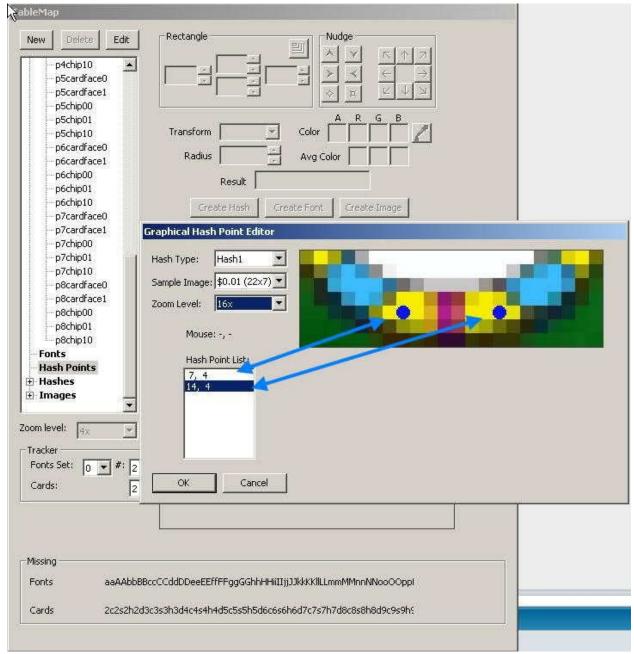


We will use hash points for recognizing chips, but not hash0 which is already in use, but this time hash1 to avoid potential hash-conflicts, even if they are very unlikely. We create these hash points with the "Graphical Hash Point Editor"

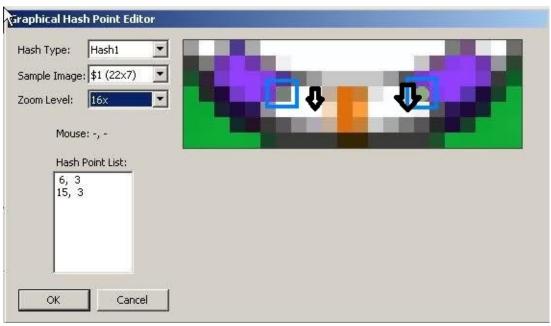
- 1. click Hash Points
- 2. click Edit
- 3. select the hash type
- 4. select one chip image
- 5. zoom to 16x



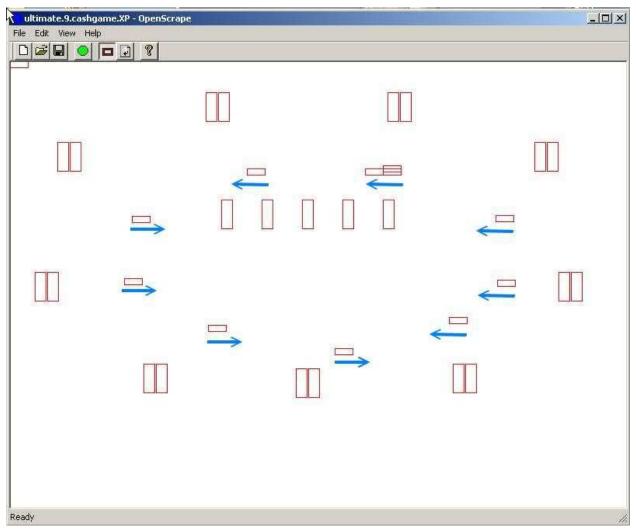
As you can see we created two new hash points of type "hash1" for this chip. Please note, that both points are at the side panel of the chip, which is always visible, even if the stack consists of multiple chips. We want OpenHoldem to be able to count them properly.



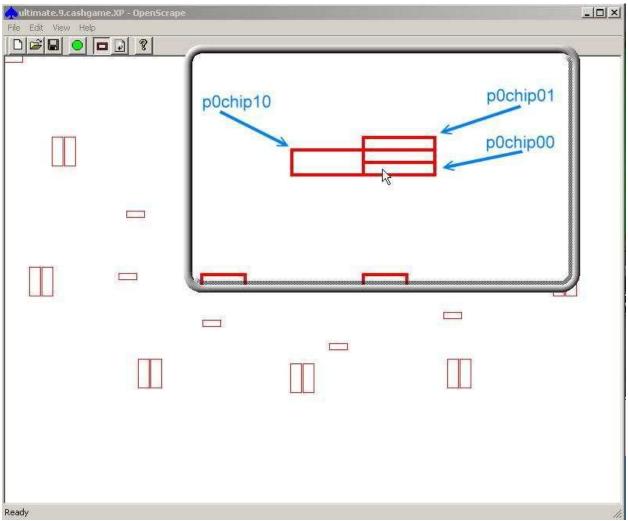
As it turned out both the \$1 chip and the \$1000 chip show the same white pixel at the formerly chosen hash-points (marked by arrows) This causes a hash collision and forces us to fine-tune the location of the points so that all chips can be properly recognized.



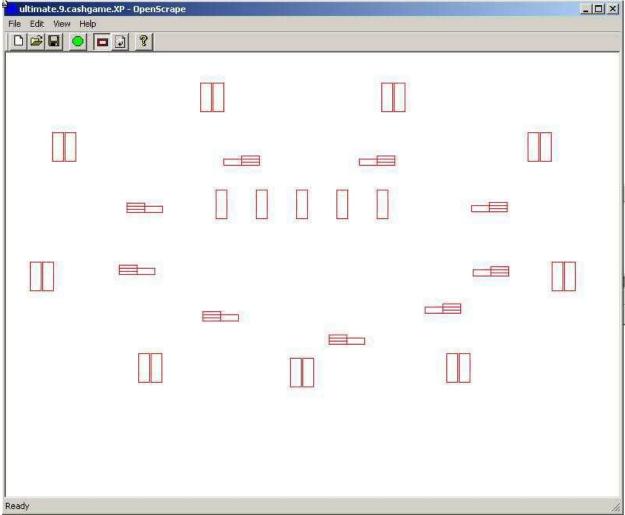
Next we have to care about the location of additional chip stacks. As you can see they can grow in different directions for different playewrs.



This determines the placement of our chip regions. pXchip00 is always the first chip, pXchip01 is always on top of that and pXchip10 is the bottom chip of the second chip stack. OpenHoldem is then able to detect all other chip-stacks and all additional chips on top automatically.

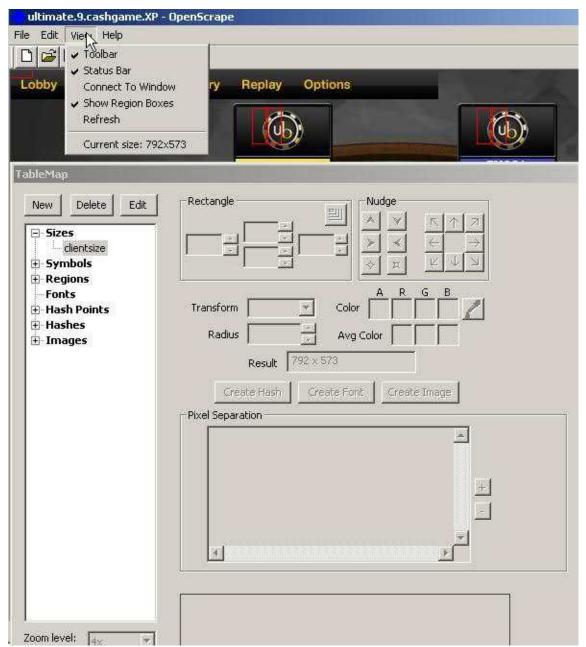


Here you can see the pXchipYZ aligned in OpenScrape.

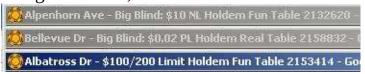


Now we have a nearly working tablemap. It recognizes cards and bets. Names are optional and only necessary for PokerTracker and other kinds of opponent modelling. The balances are also necessary, they can be done with font-scraping, which will be discussed in another chapter. We are nearly finished now, but to make use of this fine map we have to tell OpenHoldem how to connect to a table. This happens by table-size and title-string (and in some rare cases by optional table-points, if different game-types (6max / 10max) can't be played with the same map).

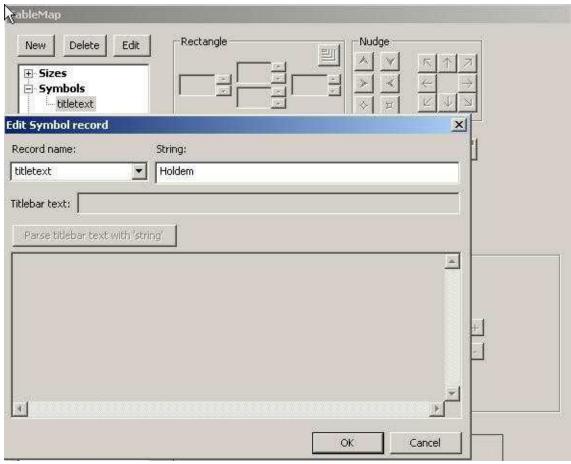
So first we have to define the symbol *clientsize*. The necessary value gets provided by OpenScrape.



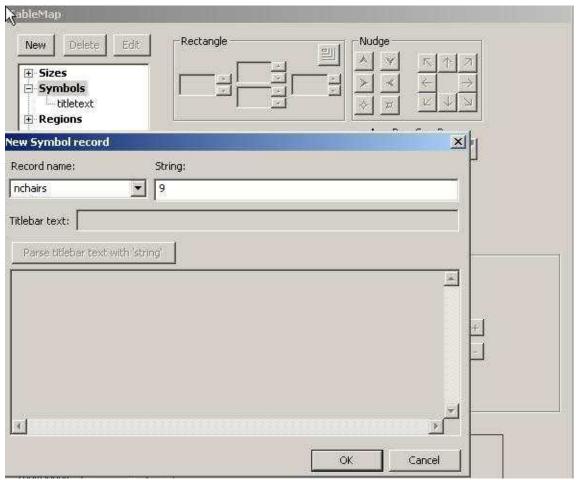
Next we have to define a titletext. It should be unique for each casino (at least if other casinos have the same table size), but it should match all game-types that are supported by our map. Our map supports NL, FL and PL, which share the string "Holdem", so we choose this one as a title identifier.



The *titletext* is a symbol-record that can be defined easily with the tablemap editor.



We also have to tell the scraper how many chairs the tablemap supports.



As you can see in the title OpenHoldem is connected to the table (and the map would work if we also defined which chair is seated).



Now we also need common pots (this includes side-pots). They work similar to bets and chip-stacks. Here c0potXchip00, c0potXchip01 and c0potXchip10.



Most sites have three pots, but Ultimatebet has five pots (the fifth pot does not appear in the situation below, but we show its location)

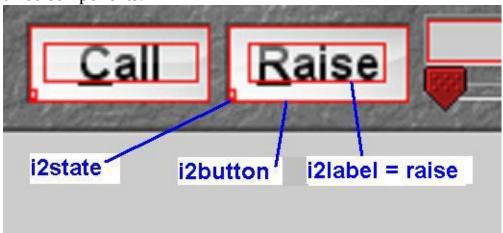


That's it — basically.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-20T20:58:01.156000

How the Interface-Buttons work

The holdem scraper has the concept of interface regions, that can be clicked by the autoplayer to interact with the casino. Such an interface region consists of three components:



Location of the button, where X is the buttons number (0..9)

iXbutton and the location is a rectangle, specified by the coordinates

of the top-left and bottom-right corner.

Visibility of the button. If the button is visible it can be

iXstate clicked by the autoplayer. If not then the action will be

skipped or a backup-action will be chosen.

iXlabel Type of the button, e.g.fold / call / rais

In the tablemap file it looks like that:

r\$i0button	491	320	502	331	000000	0 N
r\$i0state	491	331	491	331	ffffff	0 C
r\$i0label	493	322	499	330	ffffff	0 H0

The purpose of the button region is simply to tell the autoplayer where it needs to click the mouse. You can set the transform to "none", as only the position of the clickable region is of interest for the autoplayer. In the example above the position for button 0 is ((491, 320), (502, 331)).

The purpose of the state region is to determine whether or not the button is visible or not. It's generally a small color region which has to return true when the button is visible. But you can use hash as well. If you use hash, a true value will be identified as follows (so rename your images accordingly):

- yes
- checked
- true
- lit

OpenHoldem will search for any of these substrings, so "checked7" would work too, but any other string will be seen as false. Finally each interface should also have a label region or regions that will further define the purpose of the button. So the engine recognizes the following key words within labels:

- fold
- check
- call
- bet
- raise
- betsize (and "swag" for backward compatibility)
- allin (or all-in)
- prefold
- autopost (or auto-post / autoblind / auto-blind / postblind / post-blind)
- sitin (or sit-in)
- sitout (or sit-out)
- leave (or exit)

...and various forms of nearly fitting scrapes (e.g. "aut0post", "a111n" and other kind of leetspeak-lookalike). Any iXlabel region that produces one of these keywords will be seen by the engine as clickable for that specific action. Also all the strings starting with these keywords will work: examples: "calllll", "check01", "Raiseto:\$10" etc.

Sitin/Sitout particular behaviour



Traditionally since the early days of FullTilt some special interface-buttons are check-boxes: sitin / sitout / prefold / autopost blinds. These buttons work slightly differently than the normal action buttons. Here a button-state of "yes" or "true" means the button is checked, whereas "no" or "false" means it is unchecked and can be clicked. This is a Winholdem legacy, and derives from the first sites supported, which used check-boxes as most sites still do today. Therefore the bevaiour is inversted (compared to normal action-buttons) and these buttons will be seen as clickable if their state is "off".

The spam buttons

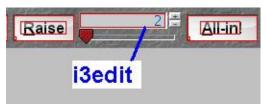


OpenHoldem provides some special buttons for spam handling i86 and if you need more i860...i869.. For spam we intend all the popups that can occlude your table. Some sites advertise tournaments about to start, other ask for confirmation when you leave the table, etc. OH can manage these popups and close them as long as they appear inside the poker table in a consistent position. There's no label region to scrape here, as type of the button is "spam" by default; just state and button matter. The autoplayer will click them automatically as soon as they become visible.

- 1. Shoot a frame of the table with the popup showing, and create i86state as a color or hash region which returns true like you would for any other button.
- 2. Paint a button region where you want OH to click to kill the popup. Done.

Notice that this is evaluated every heartbeat, so OH is very fast at closing these popups. If popups show in inconsistent locations or out of poker table, you need an external script, or external software like PTFB to manage them.

The betsize interface



If we play NoLimit or PotLimit games we need a way to enter betsizes. This is called "betsizing" (formerly "swagging", WinHoldems abbreviation for "stated wager"). To make betsizing work you need to configure several things:

- first you need an input-box, which is *i3edit*, that specifies the region where the autoplayer should enter the betsize
- before we can enter the text into i3edit we should delete the old one or the default one. To do this properly we have to define how to select and how to delete this text.
 - betsizeselectionmethod, the supported methods for selection are: Click Drag / Dbl Click / Nothing / Sgl Click / Triple Click
 - betsizedeletionmethod, the supported methods for deletion are: Delete
 / Backspace / Nothing
- once the i3edit-field got cleared the autoplayer can enter the new bet and confirm it. To confirm the bet two methods are supported (betsize*confirmationmethod*):
 - Enter
 - *Click Bet*, which clicks the i3-button, which usually is the same as "raise".

This is basically how NL-betsizing works. Every casino is a little bit different. You have to play around a bit and identify which sequence works, then you can specify the usual i3edit, i3state, i3button, i3label and the methods for selection / deletion / confirmation. And that's it — nearly.

Interpreting the betsize

Let's assume the following situation: you sit in the big blind and had to pay \$10. Everybody folds to the button who raises to \$30. Now it is your turn again and you decide to enter "90" into the raise-box. What does this mean? Well, it depends on the casino.

- at some casinos \$90 will be your final betsize (betsizeinterpretationmethod = 3). The entered amount will be identical to OpenHoldems f\$betsize function or to the RaiseTo-amount of OpenPPL.
- other casinos will add \$90 to the \$10 you already posted, so the final betsize will be \$100.(betsizeinterpretationmethod = 2)
- and finally some other casinos will add these \$90 to the \$30 of the last raiser, so the final betsize will be \$120 (betsizeinterpretationmethod = 1, RaiseBy in OpenPPL)

Again you have to play a bit with your casino and have to find out how it works, so that you can tell the autoplayer how it is expected to execute your desired action. But that's it finally — we promise!

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-10-06T14:26:22.312000

Interpreting Titletext

Most casinos show very valuable info in the titlebar, for example

Tournament 12345678 Table 5 - No Limit Hold'em - Blinds \$600/\$1200 A

Valuable informations in the example above (PokerStars) are:

• the game-type: tournament (and no-limit)

• the size of the blinds: \$600 / \$1200

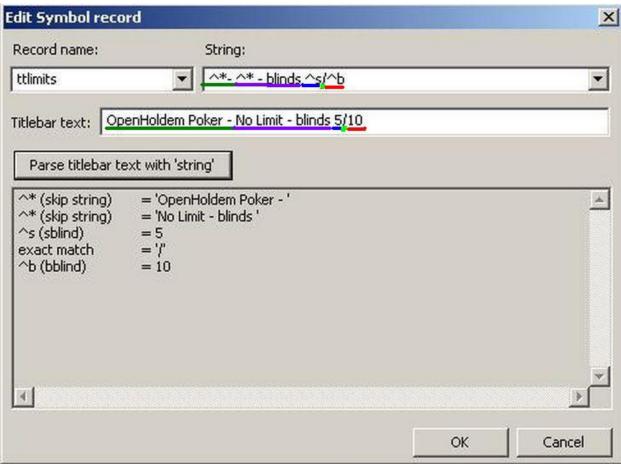
• the ante: \$75

Besides of valuable info the titlestring comes with some technical benefits:

• it is usually very reliable

• it can be accessed easily with the Windows API (GetWindowText) and does not require any scraping.

Therefore OpenHoldem provides a way to interpret the titletext:



In the example above

- "^*" followed by the constant string "- " recognized and discards everything until it finds "- " in the titlestring. This means the title-parser will skip "OpenHoldem Poker " because this data does not provide any useful info for us.
- Again there follows a dummy that needs to be skipped. "^* blinds" will ignore everything until it finds " blinds".
- Now comes the first interesting piece of data. "^s" expects a numerical value, here "5" that will be interpreted as the size of the small blind.
- "/" is a constant string that will be ignored. In our exaple it separates small-blind from big blind.
- Finally "^b" works similar like "^s": it expects a numerical value too and interprets it as the size of the big-blind.

Expert Tricks

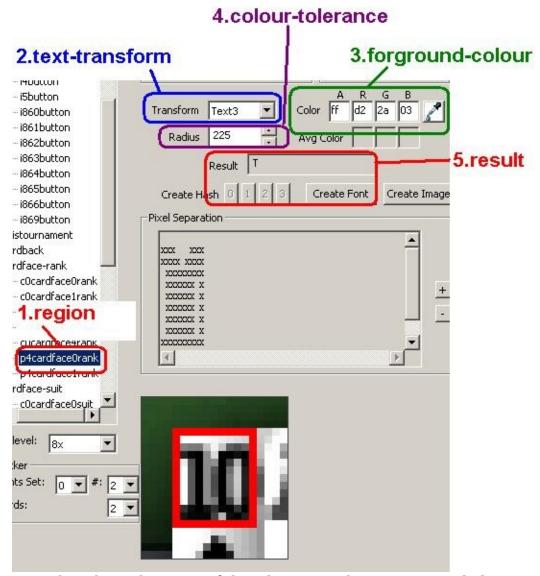
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T17:16:21.640000

Scraping "10" as Ten

When it comes to text-scraping cards most things work intuitively the natural way:

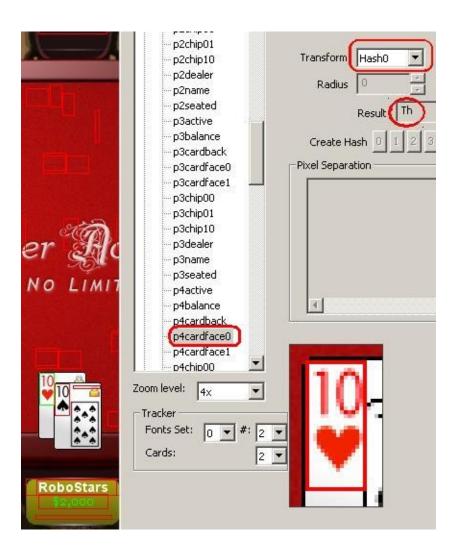
- we select a region that shows the rank of the card
- we select a transformation (here text3, but it might also be a hash or an image if there is no fuzzyness)
- we select a foreground-colour with the colour-picker and define a reasonable colour-tolerance. Sometimes we take the background-colour and a negative radius if we want to accept everything except the background. This is especially useful for blinking fonts on a non-blinking background. The colour-tolerance, called "radius", has to be in the range [0..255=0xff], where 0 requires exact identity and 255 accepts everything. (You might notice the radius of 225 below, which is near the upper end of the range. This is because the probably inexperienced creator of this tablemap chose a bad foreground-colour (the dirty red (d2, 2a, 03) at the bottom left of "1") but he wanted to accept black (0, 0, 0) which differs significantly. This only works because the small blue component (02) of his chosen foreground-colour differs from the white background colour (ff, ff, ff) by more than 225 (0xff = 255)). Usually a radius of 10 or 20 is perfect to accept a slightly fuzzy fore-ground-colour perfectly.
- if the rank of the card does not get recognized we create a new font.

And that's it!



But what about the topic of this chapter? What is so special about scraping the ten?

Well — nothing — except that some people try to create fonts for both "1" and "0", but the scraper expects a single character for a card rank: "2"..."9" for the cards of lower value, "J", "Q", "K" and "A" for the face-cards and "T" for "ten". That's the whole "difficulty" here. As you can see in the picture above we did it right, scraping the rank of ten as "T". In the picture below we do the same work, but there we scrape rank and suit together as "p4cardface0" and the result is "Th".



Scraping Dollars and Cents

For s\$ttlimits, s\$ttlimitsX, r\$c0limits/s\$c0limits, and r\$c0limitsX/s\$c0limitsX regions:

- the dollar sign (\$) leader is recognized and discarded. Commas are recognized and discarded.
- the remaining text is then used to generate a floating point number -
- the cent sign (¢) (*not the letter 'c') trailer is recognized and causes the number to be divided by 100.
- the thousand (k) and million (m) trailers are recognized and causes the number to be multiplied by 1000 or 1000000 respectively



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-02T10:09:10.546000

Scraping non-standard Titles

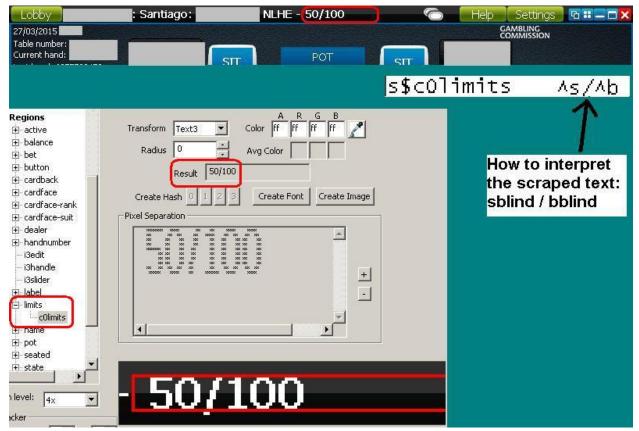
Usually poker-tables are standard-windows. They have a title that can be accessed by OpenHoldem with the Windows API. These titles contain useful information like the blind-levels and we only have to tell OpenHoldem how to interpret these titles as explained in the chapter "Interpreting Titletext".

Mold'em Cocholate NL \$1/\$2

However there are a some (very few) casinos that don't provide such a title:

- some casinos are browser-based and don't change the brwoser-title
- some poker-tables are technically based on dialog-boxes that don't posses a title.
- some casinos use standard windows with titles, but don't use them properly.

An example of the latter case are some IPoker-skins like William-Hill. They use standard windows with titles that work fine for cash-games. However when it comes to tournaments they don't update this title properly. Instead the casinosoftware over-paints the real title with some nice graphics that looks like a title, and the "real" title is frozen at the first blind-level.



Luckily there is a solution: if OpenHoldem can't interpret the title (ttlimits) it will try to scrape the blinds and all other info from the table. To do so we need:

dummy ttlimits that discards everything

s\$ttlimits ^- tourney

a region r\$c0limits that defines where to look for the painted info

r\$c0limits 375 7 474 16 ffffffff 0 T3

As you can see this region is near the top of the table (y-coordinates 7..16), horizontally near the middle (375..474) of a window with size 800x600. We use a T3-text-transform and the fore-ground-colous is white 0xFFFFFFF with a radius of 0 (exact match).

- now we need to scrape the text properly (see the chapter about "Collecting Fonts)" and we have the same info that we would get at other casinos automatically without any effort.
- finally we have to tell OpenHoldem how to interpret this text and create a symbol s\$c0limits that works exactly the same way like ttlimits.

That's all to make it work.



Finally in some very rare cases it could be that that positions and the texts for the blind-levels change, especially if you want to create a map that supports multiple game-types like in the image above. That's no problem at all, you can introduce some more regions r\$c0limits0..r\$c0limit9 and corresponding symbols s\$c0limitsX to scrape the info from the table. And if the data is spread all over the table you can scrape each snippet of information individually with c0sblind, c0bblind, c0bbet, c0ante and some more.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-04T09:52:11.321000

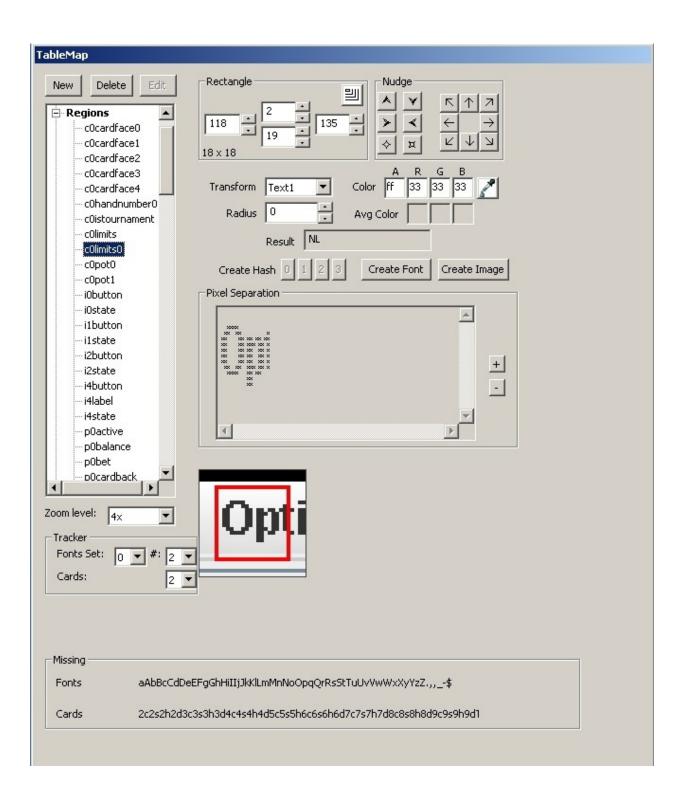
Setting The Game-Type

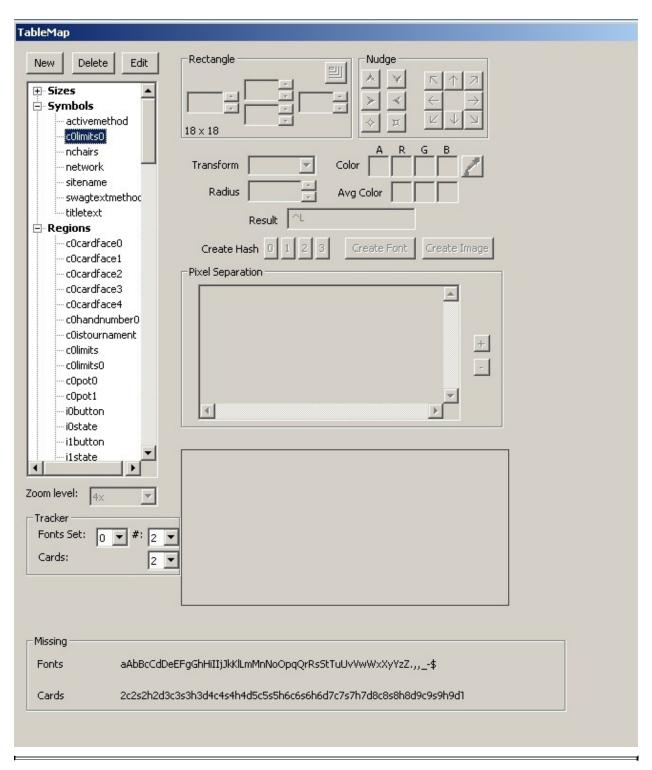
If you create a tablemap it is pretty important to tell OpenHoldem which gametype you are playing. If you play fixed limit the mimimum bet will double up at the turn. So if you play a \$1/2-game the blinds will be \$0.50 and \$1. The small bet, as it is called, will be \$1. That means your mimimum bet and all raises have to be at least 1 dollar. But the pot will grow, and betting \$1 into a large pot doesn't make much sense. So at the turn the betsizes double up and the now so-called big-bet would be \$2 in our example. This results in a \$0.50/\$1/\$2 betting-structure, contrary to the now famous no-limit with a \$0.50/\$1/\$1-structure, where you can choose your betsize as you like and the minimum bet at later streets would be the same as preflop and at the flop.

Why this elementary introduction? Because some players nowadays don't even know about the old days where fixed-limit was the only game of choice in the internet. But does that matter and how does it affect players who want to play only no-limit?

Well, per default OpenHoldem will assume a no-limit-game nowadays and usually you can get the game-type from the title-sting ("^L" in "ttlimits"). But sometimes there might be no title (or no reliable title) and you might need to define another gametype as the default one. As mostly there is an alternative way: this information can be scraped, just like anything else that matters. Using OpenScrape:

- 1. Create the region "c0limits0".
- 2. Optimally you want to choose a region at the table that displays "No-Limit", but if that is not present you might choose any fake-region that doesn't change, like "Options". Create letters that return NL for whatever is shown in the region. Use a font that doesn't get used for anything else to avoid conflicts (I used text1). I created a small region over the options field and instead of teaching OH the letter 'O' and 'p', I told it it stands for 'N' and 'L' because I want a no limit map.
- 3. Under "symbols" create "c0limits0". The symbol "s\$c0limits" simply tells OpenHoldem how to interpret the value of the region "r\$c0limits" and works exactly the same way like "ttlimits" works for the title-string. Here we set s\$c0limits to "^L" which is the gametype, as advertised in the section about symbol-records..
- 4. Live happily ever after.



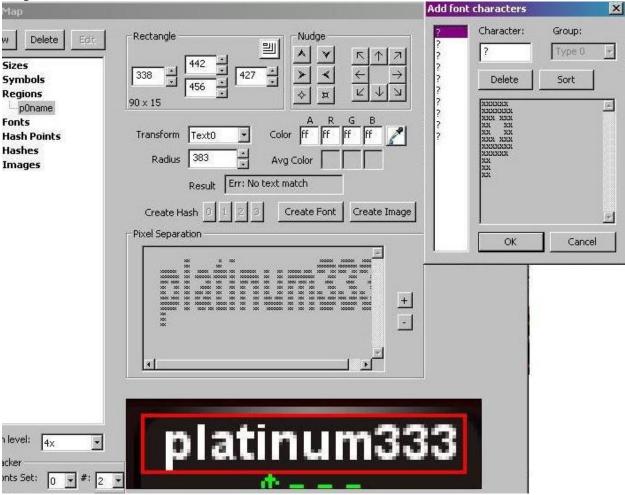


Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T17:23:23.265000

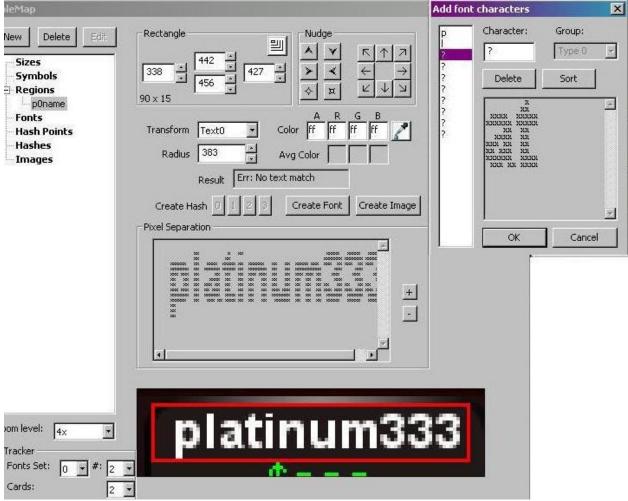
Scraping fonts which are glued together

This chapter is about font creation, precisely about the main difficulty that can happen: sometime two or more characters seem to touch each other, they are printed without any visible distance. This can cause problems because the scraper tries to recognize one character after another, but out of the box it can't handle groups of characters that look like something unknown.

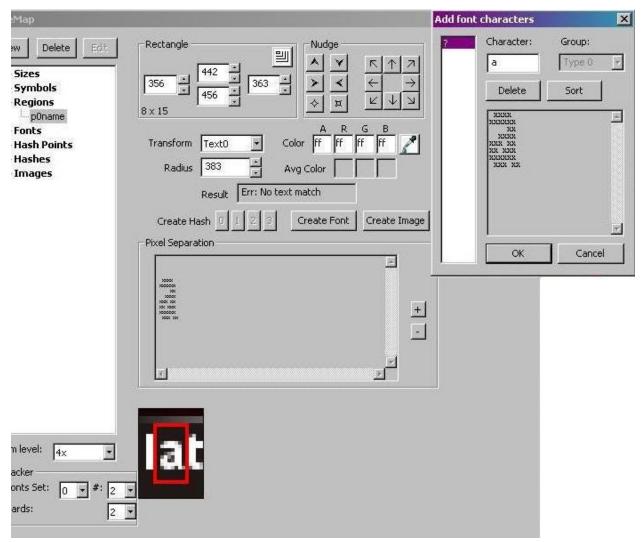
At the first picture below you see the good case: OpenScrape tries to recognize the playername "platinum333" and the first character "p" gets properly separated from the rest. All we have to do here: tell OpenScrape that this group of pixels is a "p".



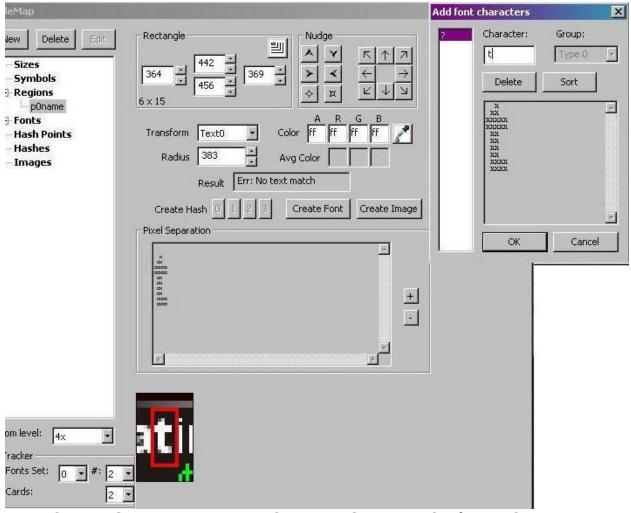
The second character "l" works also properly, but the next two characters are glued together: they can't be separated by a vertical empty line of pixels. OpenScrape would like to know what "at" means in the alphabet..



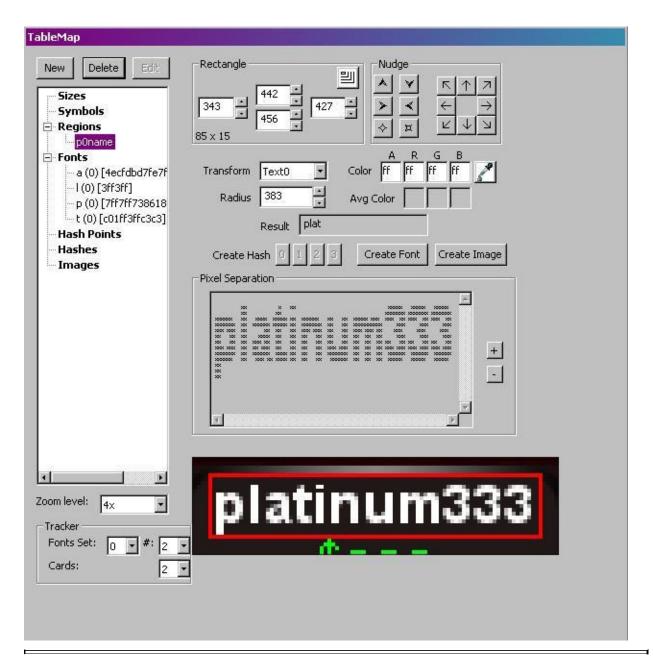
Unfortunatelly "at" is no proper character, but there is a little trick to separate these two characters. We temporary make the boundaries of the name-region smaller so that it only contains the first of these two characters — the "a". Then OpenScrape will try to recognize all characters in this small region, which contains only the pixels of "a". Now we can define "a" just like any other normal font.



We repeat this process for the second critical character "t". Then both of them can be recognized individually.



Once this got done we can restore the original region. The first 4 characters get now recognized properly and we can continue with the rest.



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:13:30.140000

Scraping blinking fonts and name-plates

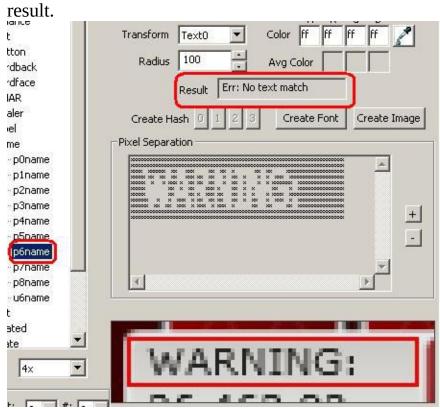
Sometimes it might be that a name-plate blinks and both the foreground and the background colour change. Then it is usually not possible to scrape the data the normal way just with one colour and one radius (colour tolerance).



But as usual OpenHoldem provides a solution for blinking fonts: a second region uXname in addition to pXname to scrape the playername. If one region fails to return good data OpenHoldem will use the other one and vice versa. In practice it looks like that: the normal name-plate (here for player 6) gets recognized perfectly by p6name.



But when the name is highlighted with inverted colours the region p6name fails to recognize the name. Please notice the inverted pixel separation (that might produce even complete nonsense if the colours were different) and the missing



But then u6name works



And together both regions will always produce a good result.

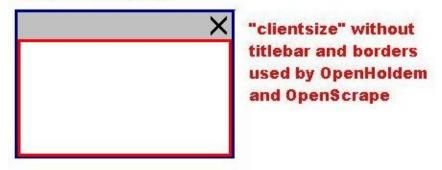
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-26T07:35:58.015000

Size Records

Description

Size records are used for one purpose only, to allow OpenHoldem's autoconnector to find windows of the correct size. More specifically, they provide the ability to specify the accepted sizes for the casino table that the scraper engine will recognize.

total window size

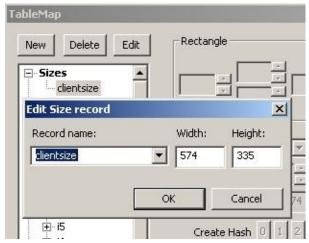


Upon connection OpenHoldem will look for poker table windows that are between the *clientsizemin* and *clientsizemax* sizes. Thereafter it will use *targetsize* to resize the table one time after connection to the preferred size (if resizing is needed at all).

Manipulating size records using OpenScrape

To create a new size record, select the "Sizes" category in the record tree display on the Table Map Editor window, and click the "New" button. A window will be displayed that allows you to enter the width and height parameters for this new record. To edit an existing size record, select the record in the record tree display on the Table Map Editor window, and click the "Edit" button. A window will be displayed that allows you to adjust the width and height parameters for this record, or change the record name itself. To delete an existing size record, select the record in the record tree display on the Table Map Editor window, and click the "Delete" button. A window will be displayed asking for verification of this action.

A note about window client sizes: Microsoft Windows provides several different measures of a windows' size, such as the complete window including frame and title bar, and an altogether different measure, the so-called "client area", that lies inside the window frame and excludes frames and title bars. OpenHoldem references the Microsoft "client area" measurement for these "Size" records. To help you understand the size of the poker window that you have attached to (with the Green Circle Button), OpenScrape always displays the client size of the attached poker table in the "View" menu on the main OpenScrape window. The menu option is at the bottom of this menu and is preceded by "Current Size: ". The width and height displayed in this menu can be entered directly into a clientsize record.



Technical reference

In the Table Map (.tm) file, the size records will be preceded with the characters "z\$". These records have the following format:

z\$<name> <width> <height>

<name> is the name of the record (see table below) <width> and <height> are integer values.

Valid size records and their descriptions:

Size record	Description
clientsizemin	Specifies the minimum allowed client size of the poker table window.
clientsizemax	Specifies the maximum allowed client size of the poker table window.
targetsize	Specifies the optimal size of the poker table window. If this parameter gets used the table will be resized once after connection to match the perfect size.

Region Records

Description

The purpose of the Region records is to describe named rectangular areas on a poker table, each of which contains an element of the game state. These regions will "return" values to the OpenHoldem game state engine, depending on the type of transform selected.

An example will make this clearer. Each seat at the table will have two cards in front it, and we want our screen scraper to tell the OpenHoldem game state engine what those cards are. If the casino client displays a group of pixels for our first hole card, that to a human being looks like an ace of hearts, we need somehow to tell the screen scraper to "return" "Ah" to the game state engine each time that group of pixels is seen in one of the card locations. To do this we collect an image for that ace of hearts, then setup a region record that returns "Ah" to OpenHoldem when that ace of hearts is seen during real live game play.

Manipulating region records using OpenScrape

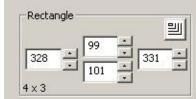
To create a new region record, select the "Regions" category in the record tree display on the Table Map Editor window, and click the "New" button. A window will be displayed that allows you to select the name of the region to create. After creating the record, you can use the controls on the right side of the Table Map Editor window, as described below, to change the settings for the new record. To edit an existing region record, select the record in the record tree display on the Table Map Editor window. Immediately after selecting the record, the controls on the right side of the Table Map Editor window are updated with the settings for the selected region record. These can then be changed, as described below. To delete an existing region record, select the record in the record tree display on the Table Map Editor window, and click the "Delete" button. A window will be displayed asking for verification of this action.

Note: The Insert and Delete keys can also be used as a shortcut for adding and deleting Region records.

Note: Regions can also be selected by left clicking within the region's flashing red rectangle that is layered on top of the poker table image on the Main window. If multiple regions are layered on top of each other, clicking again on the same area will select the next region below the selected region. This only

works for two regions. If three or more regions are layered on top of each other, the selection tree will need to be used.

The majority of the right side of the Table Map Editor window is used to manipulate the parameters of the region record.



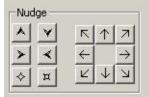
Starting at the top, there is a rectangle group box. This group box allows you to define the extents (left, right, top, bottom) of the region record's bounding rectangle. The bounds are inclusive – in other words, a rectangle defined as left=0, right=5, top=0 and bottom=6 will be 6 pixels wide and 7 pixels high. The rectangle settings can be direct entered into the controls, you can use the spinners next to the controls, or you can use the rectangle drawing tool to quickly draw the rectangle on the poker table.

To use the rectangle drawing tool, click once on this control:



...then move your mouse cursor to the Main OpenScrape window where the image of the poker table is displayed. The mouse cursor will change to look like crosshairs with an arrow pointing right and down. Left click and hold the mouse button anywhere on the poker table image to define the left-top corner of the rectangle. Drag to the right and down, and when you reach the right-bottom corner of the rectangle, release the left mouse button.

A zoomed view of the region is displayed in the box at the bottom right of the Table Map Editor window. The Table Map Editor window can be enlarged, if needed, to show the entire region. Use the Zoom Level control to adjust the zoom level in this display.



To the right of the rectangle group box is the nudge group box. These controls allow you to adjust the size and position of the region record's rectangle in a variety of ways. The group of six controls on the left, starting from top left and

moving clockwise, allow you to make the rectangle taller, shorter, thinner, smaller, bigger and wider. The group of 8 controls on the right, starting from top left and moving clockwise, allow you to move the rectangle up-left, up, up-right, right, down-right, down, down-left, and left.

Note: Regions can also be moved by first selecting the region by left clicking within the region's flashing red rectangle on the poker table image on the Main window, then by holding the Shift key and dragging and dropping the region to its new location.

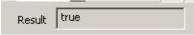
Keyboard shortcuts to move and resize regions

- Arrow keys Move the region 1 pixel
- Numpad keys 1, 3, 7, and 9 Move the region diagonally 1 pixel
- Shift + Arrow keys Move the region 5 pixels
- Control + Arrow keys Resize the region by 1 pixel
- Control + Shift + Arrow keys Resize the region by 5 pixels



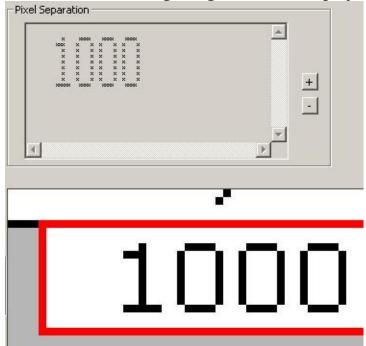
The next set of controls allows you to set the remainder of the parameters on a region record. Here you can set the type of transform, the radius of the RGB color cube, and the center color of the RGB color cube. The use of RGB color cubes for transforms of type Color and Text is described in the technical reference below. Also in this group of controls is an eyedropper button that is used to populate the color parameters, as opposed to directly entering those values. To use the eyedropper, click once on the eyedropper button, then hover over the region's image display at the bottom of the Table Map Editor window. When the color you want is present in the four color fields, click the left mouse button to lock them in.

As a helpful reference, the "Avg Color" fields are populated with the calculated average color of all pixels in the region. This is useful for finding RGB color cube centers and for text transform separations.



This field provides an instant preview of the results of combining the current pixels in the region with the current transform for that region. If this is a Text transform, the actual transformed ASCII text is displayed. If this is a Color transform, True or False is displayed, depending if the region's average color is

inside or outside of the defined RGB color cube. If this is a Hash transform, the value of the matching Hash record is displayed. If this is an Image transform, the value of the matching Image record is displayed.



If your region is defined to have a Text transform, then the Pixel Separation window will be populated with a preview of this separation. The key to good Text transformation is the separation of foreground pixel colors from background pixel colors. In this display, foreground pixels will be shown as an "X", and background pixels as a blank space. The size of the font used in this display can be adjusted with the "+" and "-" buttons to the right. Note: Remember that the whole purpose of a region definition, with a given type of transformation, is to tell the screen scraper what the pixels mean that are displayed on the screen within that region. The return value of the transformation is provided to the OpenHoldem game state engine which then uses those values to establish the game state and to provide various information that your bot logic needs to make action decisions. A human can easily look at the balance for seat #2 and see that it is \$950.12. For a computer it is harder – we need to specify the rectangle that bounds this balance region, define it as a text transform, choose a color that lets the screen scraper identify foreground from background pixels, and then collect each of the individual font characters (more on this below) so that the screen scraper engine can look up the actual ASCII text for these pixels.

The Edit menu of on the Main window contains a "Duplicate Region" entry, which is intended to allow the quick creation of related region records. First,

select a region record from the tree display on the Table Map Editor window, then select Edit/Duplicate Region from the Main window. This action will display this window: In this case, the p0balance region was selected. You are given the option to duplicate the selected region record to other region records of the same type. In this case, all of the unpopulated pXbalance and uXbalance region records are candidates. You can select one or more than one destination record here (using the standard Windows Ctrl and Shift multi-select capability) to copy the selected region to. When you click the OK button, the new region records will be created for you, and you can then move them where needed and adjust them appropriately.

Technical Reference

In the Table Map (.tm) file, these records will be preceded with the characters "r\$". Region records have the following format:

r\$<name> <left> <top> <right> <bottom> <color> <radius> <transform> <name> is the name of the record (see table below) <left>, <top>, <right> and <bottom> are integer values within the casino table, where the upper left corner of the client area is considered to be 0, 0 <color> is a 1-8 character hexadecimal number in the form AABBGGRR (Microsoft Window's COLORREF format (http://www.w3schools.com/colors/colors_names.asp)). The usage of the color value is dependent on the transform selected. <radius> is an RGB color distance. <transform> describes the engine algorithm that should be used to interpret the pixels in the region.

Valid Transforms And Their Desciption

Transform Description

This is used for those regions that do not need a transform as their only purpose is to describe a rectangle for use by the scraper engine. An example is button location (i0button - iFbutton) which informs the Autoplayer engine of the allowed rectangle for mouse clicks.

This is a calculation of the average RGB color for the entire region. Used in conjunction with the radius, the engine will determine if the contents of the region, as seen during game play, match the color/radius as specified in the Table Map. The color value in the region record defines the center of a cube, and the radius defines

C (color)

N (none)

the dimensions of that cube. This cube is then mapped to an RGB color space, and if the average color of the region falls within this cube, then a match is reported. A negative radius can also be specified, in which case if the average color of the region falls outside of the cube, then a match is reported.

This transform does two things. It first searches the Image records in the Table Map for a pixel-by-pixel match, and returns the value of that Image record if a match is found. If no pixel-by-pixel match is found, the transform then searches for the closest match, based on an perceptual difference algorithm (http://pdiff.sourceforge.net/

I (image)

). There is an internal hard-coded limit, which prevents Image records from matching the region if 65% or more of available pixels of that region do not match. If this limit did not exist, then clearly non-matching regions could return all sorts of junk. This exact-match-to-pdiff failover is automatic, and requires no additional work from the Table Map creator other than to set the transform to "I" and to collect the images. This is a powerful transform, but be aware that this power comes at the cost of CPU cycles.

This is an OCR-algorithm (

http://en.wikipedia.org/wiki/Optical_character_recognition

T (text)

) for the purpose of converting screen pixels to ASCII text. The color cube specifications (as in the C transform) are used to separate the foreground from the background pixels. A lookup is then performed against the Font records in the Table Map on a character by character basis to produce a resultant text string. This transform reduces the contents of the rectangle to a 32-bit value. OpenHoldem and OpenScrape use only a single, well respected hash algorithm that is open for public review. Because of the selection of this algorithm, we have not yet seen a hash collision (that did not have an unrelated root cause, like record duplication). That is not to say that a hash collision is not impossible, however the mapping of 52 card images (for example) into a 32-bit address space makes this highly unlikely. If a collision ever occurs, there is

the option to use hash types other than Type 0. Types 0-3 all use different hash seeds with the same algorithm, and thus if a collision

Hn (hash n) is seen, one could simply change the hash type for that region.

The Type 0 hash algorithm uses all pixels in a region to calculate a hash. To use specific pixels in the region, use Types 1-3. Types 1-3 default to using all pixels in a region as well, but if Hash Point records are present in the Table Map and of a matching type, the engine will use those specific points for calculation of the hash. Once a region is hashed, the resultant 32-bit hash value is then looked up in the h\$ records in the Table Map. If a match is found, the name of the h\$ record is returned and acted on appropriately (i.e. common card 1 is Queen-Hearts, Fold button is present, Player 2 is seated, etc.)

This transform takes the pixel in the middle of the region and W (colour) transforms it to the closest web-safe colour (216 possible values). It is meant for player-profiling with colourcodes.

Valid region records and their descriptions

Record Description

Common cards 0-4; represents the entire card, the best transformation is probably hash. Typically either c0cardfaceX is used, or c0cardfaceXrank / c0cardfaceXsuit, but not both. Valid results are a case-insensitive, two character string. The first character is the rank and should be one of: "2", "3", "4", "5", "6", "7", "8", "9", "t", "j", "q", "k", or "a". The second character is the suit and should be one of: "c", "d", "h" or "s". c0cardface0rank - c0cardface4rank Rank of common cards 0-4; the best transformation is probably text. Typically either c0cardfaceX is used, or c0cardfaceXrank / c0cardfaceXsuit, but not both. Valid results are case-insensitive, and are one of "2", "3", "4", "5", "6", "7", "8", "9", "10", "t", "j", "q", "k", or "a".

c0cardface0suit - c0cardface4suit

c0cardface0 - c0cardface4

Suit of common cards 0-4; the best transformation is probably text. Typically either c0cardfaceX is used, or c0cardfaceXrank / c0cardfaceXsuit, but not both. Valid results are case-insensitive, and are one of "c", "d", "h" or "s".

Ten regions to specify where the game state engine should find the current handnumber. The first region

c0handnumber, c0handnumber1 c0handnumber9 found with valid numeric information will be used and the remainder skipped. c0handnumber is searched first, then c0handnumber1 to c0handnumber9 in that order. The handnumber will be used internally to detect handresets (depending on handresetmethod) and is also accessible at the DLL-level. To avoid bad scrapes and invalid handresets the handnumber is optionally ranged between s\$handnumber_min_expected_digits and s\$handnumber_max_expected_digits.

c0smallblind

Identifies the region on the poker table where the small blind can be found. Title text parsing is much easier to implement if that option is available, otherwise a text transformation can be used with this region to identify the small blind.

c0bigblind

Identifies the region on the poker table where the big blind can be found. Title text parsing is much easier to implement if that option is available, otherwise a text transformation can be used with this region to identify the big blind. c0bigbet Identifies the region on the poker table where the big bet can be found. Title text parsing is much easier to implement if that option is available, otherwise a text transformation can be used with this region to identify the big bet.

c0ante

Identifies the region on the poker table where the ante can be found. Title text parsing is much easier to implement if that option is available, otherwise a text transformation can be used with this region to identify the ante.

c0pot0 - c0pot4

The value of the main pot should be identified by c0pot0 using a text transform.

c0pot1 to c0pot4

can be used to identify side pots if required. c0potNchipXY See chip scraping instructions for p0chipXY - p9chipXY below. The difference is that the "N" refers to the different pots - N=0 is the main pot, N=1-4 are side pots.

c0limits, c0limits0 -

Identifies the region on the casino table where the limit information can be found; this region is used in conjunction with the Symbol records collimits and c0limits9

collimits 0-collimit 9 text parse strings. This region should really only be used if title text parsing is not an option.

i0button - i9button, iAbutton - iFbutton

Specifies the rectangular area in which the Autoplayer can click the mouse for each of the 16 buttons.

Specifies the action that should be associated with buttons the 16 buttons. There are no defauls, so every button needs a label.

The return value from the transformation of this region

(hash or text) will determine how the scraper engine interprets the button. All values are changed to lowercase and have spaces and '-' removed. Only the leftmost characters will be tested "allin", "a11in", "allin", "a111n", "aiiin": button is seen as the allin button "raise", "ra1se", "ralse", "bet": button is seen as the raise button "call", "caii", "ca11": button is seen as the call button "check": button is seen as the check button "fold", "fo1d", "foid": button is seen as the fold button "autopost", "aut0p0st": button is seen as the autopost button "sitin", "s1t1n": button is seen as the sitin button "sitout", "s1tout", "s1t0ut", "s1t0ut": button is seen as the

Examples: "All-In" will be recognized as allin; "Autopost blinds" will be recognized as autopost

leave button "prefold": button is seen as the prefold

button

Specifies whether the corresponding button is live and available to be clicked. The best transformation is hash or color. If the leftmost characters of the result from this region's transform are either "true", "on", "yes", "checked" or "lit" then the button will be seen as available to be clicked. If any other result is returned, such as "false", then the button will be seen as not available to be clicked. Results are case-insensitive. Specifies the rectangular region that can be used for entering a betsize.

i0state - i9state.

iAstate - iFstate

Similar to the i0button - i9button regions, however these regions are intended to be used to combat spam/popups

i0label - i9label, iAlabel - iFlabel

i3edit

i860button - i869button

that could occlude the poker window. If the corresponding state region returns true, the Autoplayer will try to click in this region to dismiss the spam/popup.

The corresponding state regions for the i86button,

i860button - i869button regions above. The best transformation is hash or color. If the leftmost characters of the result from this region's transform are either "true", "on", "yes", "checked" or "lit" then the button will be seen as available to be clicked. If any other result is returned, such as "false", then the button will be seen as not available to be clicked. Results are case-

insensitive.

Used to scrape the corresponding MTT symbols from the table

mtt_number_entrants

i860state - i869state

mtt_players_remaining mtt_paid_places mtt_my_rank mtt_largest_stack mtt_average_stack

Used to determine if a player is active (sitting in) or not (sitting out). Any transform can be used equally well with this region. If the leftmost characters of the result from this region's transform are either "true" or "active", then the region is considered to be active. If any other result is returned, such as "false", "inactive", "out" or "away", then the chair is not considered to be active. Results are case-insensitive.

p0bet - p9bet

Used to identify a player's balance. It only makes sense to use a text transform for these regions.

Used to identify a player's bet. It only makes sense to

use a text transform for these regions.

p0balance - p9balance

Used to identify when a seat 0-9 displays a card back. The best transformation is hash or color. If the leftmost characters of the result from this region's transform are either "true" or "cardback", then the region is

p0active - p9active

mtt smallest stack

p0cardback - p9cardback

considered to be showing a cardback. If any other result is returned, such as "false", then the chair is not considered to have be showing a cardback. Results are case-insensitive.

p0cardface0 - p9cardface0

The first card for seats 0-9; represents the entire card. The best transformation is probably hash. Typically either pXcardfaceY is used, or pXcardfaceYrank / pXcardfaceYsuit, but not both. Valid results are a case-insensitive, two character string. The first character is the rank and should be one of: "2", "3", "4", "5", "6", "7", "8", "9", "t", "j", "q", "k", or "a". The second character is the suit and should be one of: "c", "d", "h" or "s".

p0cardface1 - p9cardface1

The second card for seats 0-9; represents the entire card. The best transformation is probably hash. Typically either pXcardfaceY is used, or pXcardfaceYrank / pXcardfaceYsuit, but not both. Valid results are a case-insensitive, two character string. The first character is the rank and should be one of: "2", "3", "4", "5", "6", "7", "8", "9", "t", "j", "q", "k", or "a". The second character is the suit and should be one of: "c", "d", "h" or "s".

The nocard-regions determine if a player holds no card at all, no card-faces and no cardbacks. They got introduced for two reasons:

p0cardface0nocard - p9cardface1nocard

- performance, because often the costly scraping of cardfaces can be avoided
- better error-detection, because now all three possibilities (no cards, card-faces and card-backs can be explicitly scraped instead of implicitly returning "no card" if the other two fail).

The best transformation is probably hash.

The rank of the first card for seats 0-9. The best transformation is probably text. Typically either pXcardfaceY is used, or pXcardfaceYrank /

p0cardface0rank -

p9cardface0rank

pXcardfaceYsuit, but not both. Valid results are case-insensitive, and are one of "2", "3", "4", "5", "6", "7", "8", "9", "10", "t", "j", "q", "k", or "a".

p0cardface0suit p9cardface0suit The suit of the first card for seats 0-9. The best transformation is probably text. Typically either pXcardfaceY is used, or pXcardfaceYrank / pXcardfaceYsuit, but not both. Valid results are case-insensitive, and are one of "c", "d", "h" or "s".

p0cardface1rank p9cardface1rank The rank of the second card for seats 0-9. The best transformation is probably text. Typically either pXcardfaceY is used, or pXcardfaceYrank / pXcardfaceYsuit, but not both. Valid results are case-insensitive, and are one of "2", "3", "4", "5", "6", "7", "8", "9", "10", "t", "j", "q", "k", or "a". p0cardface1suit - p9cardface1suit The suit of the second card for seats 0-9. The best transformation is probably text. Typically either pXcardfaceY is used, or pXcardfaceYrank / pXcardfaceYsuit, but not both. Valid results are case-insensitive, and are one of "c", "d", "h" or "s".

p0colourcode p9colourcode The colourcode for players 0..9, meant for easy opponent profiling

p0dealer - p9dealer

Identifies if the dealer button is found for seat 0-9, or not. Color transforms are usually used for these regions, but hash, text or image transforms can be used too. If the leftmost characters of the result from this region's transform are either "true" or "dealer", then the chair is considered to have the dealer button. If any other result is returned, such as "false", then the chair is not considered to have the dealer button. Results are case-insensitive.

p0name - p9name

Identifies the name of the player in seat 0-9. It only makes sense to use a text transform for these regions. Identifies whether seat 0-9 is occupied (seated) or not. Color transforms are usually used for these regions, but hash, text or image transforms can be used too. If the leftmost characters of the result from this region's transform are either "true" or "seated", then the chair is considered to be seated. If any other result is returned,

p0seated - p9seated

such as "false" or "unseated", then the chair is not considered to be seated. Results are case-insensitive.

Note: Chip scraping should *only* be used if the actual numeric bet text cannot be found elsewhere on the poker table. Text transformation is magnitudes of effort easier to accomplish.

Step 1 : Ensure that the poker site uses static landings for all of the chip stacks. Step 2: Determine the size of an individual chip, and should include any portion of the chip that is painted anywhere within the window regardless of how faded or transparent. Step 3 : Create regions for the initial base of each player's chip stacks with the size determined in step 2; p0chip00, p1chip00, ..., p9chip00. Step 4 : Create vertical stride chip regions for each player's chip stack; p0chip01, p1chip01, ..., p9chip01. These regions should perfectly surround the 2nd chip that rests above the base pXchip00. Step 5: Create horizontal stride chip regions for each player's p0chipXY - p9chipXY chip stack; p0chip10, p1chip10, ..., p9chip10. These regions should perfectly surround the base chip in the second stack. Step 6 : Optional and rare: If the vertical stride between chips in a stack is not uniform, then you will need to define pXchip02-pXchip09 for each chip in the stack. For those sites with non-uniform vertical chip strides, this means that there is a limit of 10 chips per stack that will be recognized by the engine. Step 7: Optional and rare: If the horizontal stride between chip stacks is not uniform, then you will need to define pXchip10-pXchip90 for each stack. For those sites with non-uniform horizontal chip strides, this means that there is a limit of 10 stacks that will be recognized by the engine. In a worst case situation, there may be 100 chip regions * 10 players = 1000 region definitions required to scrape chips! (So now go look at scraping the bet text instead)

These optional regions can be used to distinguish casinos and tables that can't be recognized by usual means, i.e. by tablesize and titletexts only. Tablepoints

tablepoint0 - tablepoint9

are usually colour-transforms that get placed at identifying regions of the table. In case tablepoints get used OpenHoldem will regularly check them again and reconnect with a different tablemap in case of a changing table-theme (final table in MTTs).

u0active - u9active

These are alternatives to the p0active-p9active regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. Any transform can be used equally well with this region. If the leftmost characters of the result from this region's transform are either "true" or "active", then the region is considered to be active. If any other result is returned, such as "false", "inactive", "out" or "away", then the chair is not considered to be active. Results are case-insensitive.

u0balance - u9balance

These are alternatives to the p0balance-p9balance regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. It only makes sense to use a text transform for these regions.

u0bet – u9bet

These are alternatives to the p0bet-p9bet regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. It only makes sense to use a text transform for these regions.

u0cardface0 u9cardface0 These are alternatives to the p0cardface0-p9cardface0 regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. Valid results are a case-insensitive, two character string. The first character is the rank and should be one of: "2", "3", "4", "5", "6", "7", "8", "9", "t", "j", "q", "k", or "a". The second character is the suit and should be one of: "c", "d", "h" or "s".

u0cardface1 - u9cardface1

These are alternatives to the p0cardface1-p9cardface1 regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. Valid results are a case-insensitive, two character string. The first character is the rank and should be one of: "2", "3", "4", "5", "6", "7", "8", "9", "t", "j", "q", "k", or "a". The second character is the suit and should be one of: "c", "d", "h" or "s".

These are alternatives to the p0dealer-p9dealer regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. Color transforms are usually used for these regions, but hash, text or image transforms can be used too. If the leftmost characters of the result from this region's transform are either "true" or "dealer", then the chair is considered to have the dealer button. If any other result is returned, such as "false", then the chair is not considered to have the dealer button. Results are case-insensitive.

These are alternatives to the p0name-p9name regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. It only makes sense to use a text transform for these regions.

These regions are alternatives to the p0seated-p9seated regions, and are used for the player's seat. They are typically used when the player's seat changes appearance when it is the player's action, to avoid misscrapes due to change in color/size, whatever. Color transforms are usually used for these regions, but hash, text or image transforms can be used too. If the leftmost characters of the result from this region's transform are either "true" or "seated", then the chair is considered to be seated. If any other result is returned, such as "false"

u0dealer – u9dealer

u0name - u9name

u0seated - u9seated

or "unseated", then the chair is not considered to be seated. Results are case-insensitive.

Symbol Records

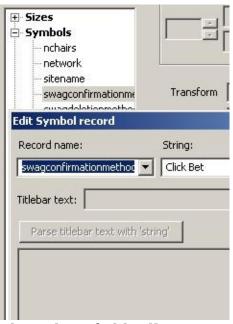
Description

Symbol records are a general purpose record type that serves many different purposes within OpenHoldem and OpenScrape. The common characteristic of these records is that they all contain a name and a free-form text field. The interpretation of that text field is different, depending on the name of the symbol record. The table in the technical reference section below describes how the free-form text is interpreted for each different Symbol record type.

Manipulating symbol records using OpenScrape

To create a new symbol record, select the "Symbols" category in the record tree display on the Table Map Editor window, and click the "New" button. A window will be displayed that allows you to enter the free-form text for this new record. To edit an existing symbol record, select the record in the record tree display on the Table Map Editor window, and click the "Edit" button. A window will be displayed that allows you to adjust the free-form text for this record, or change the record name itself. To delete an existing symbol record, select the record in the record tree display on the Table Map Editor window, and click the "Delete" button. A window will be displayed asking for verification of this action. Note: The Insert and Delete keys can also be used as a shortcut for adding and deleting Symbol records.

In the case of the "New" and "Edit" actions, the window that pops up also contains a field for "Titlebar text", a button to parse the title bar text, and a field to display the parse results. These three controls are only relevant if the record name is one of the ttlimits types. If a ttlimits record type is selected, these fields will be enabled, otherwise they will be grayed out.



These three fields allow you to test the parsing of the window's title bar text with the parse string that is specified in the "String" field, and see the results of this parse in the results field at the bottom of the window. The "Titlebar text" field is initially populated with the titlebar text of the window that OpenScrape is attached to (via the "Green Circle Button"), but this text can be overridden with anything you want to test parse.

Technical Reference

In the Table Map (.tm) file, these records will be preceded with the characters "s\$". Symbol records have the following format:

s\$<name> <text>

<name> is the name of the record (see table below) <text> is the content of the record

Valid symbol records

Symbol record Description

titletext, titletext0 - titletext9

When the "Green Circle Button" is pressed, OpenHoldem will look through each of these symbols and do a partial match of <text> to each window's title bar text. If any substring provided in any of these symbols matches, then the window is considered a match. !titletext, !titletext0 - !titletext9

When the "Green Circle Button" is pressed, OpenHoldem will look through each of these symbols and do a partial match of <text> to each window's title bar text. If all substrings provided in all of these symbols do not match, then the window is considered a match.

These symbols are intended to provide a format string that is used to parse the window's titlebar text, and extract sblind, bblind, bigbet, ante and game type information. The syntax usage of the format string is very much like the standard Clanguage format strings for scanf and printf. The placeholders use the caret character as the indicator and have the following meanings: ^h hand number ^d dummy integer (read chars 0-9 only) \(^f\) dummy float (read chars 0-9 including decimal '.') \(^s\) sblind ^b bblind ^B bigbet ^a ante ^v sb or bb depending on limit type ^V bb or BB depending on limit type ∧* skip all text up to the literal character following the * ^R roman numeral as in "Level VIII" as typically seen in SNGs ^L limit indicator For the integer and float numerics, the scanner will automatically accept and skip a leading dollar sign "\$" or leading/trailing cent sign "¢" (ascii 162 hex a2). In the case of the cent sign the target number will be divided by 100. The scanner will also recognize the standard trailing metric unit multipliers of "k" and "m" (1000 and 1000000 respectively).

The following case-insensitive strings are recognized for the ^L limit indicator: no limit, no-limit, nolimit, nl pot limit, pot-limit, potlimit, pl fixed limit, fixed-limit,

ttlimits, ttlimits0 - ttlimits9

fixedlimit, fl, limit

As an example, this title bar text "NL SNG Level VI 25/50 (2)" could be parsed with this format string "^L SNG Level ^R ^s/^b (^a)" to result in the sblind, bblind and game type being recognized.

If multiple ttlimits symbols are provided, then the engine will scan each in order, and extract matching information from the first time it is found. That means that if you specify \(^\s\) in both ttlimits and ttlimits 3, for example, it will be extracted from ttlimits and ignored in ttlimits3. OpenHoldem allows you to specify up to 11 ttlimits records in any single Table Map file. Specifying multiple ttlimits records might be needed if a given Table Map file is intended to describe multiple Texas Hold'em game flavors. However, it is a best practice to define different game flavors in different Table Maps, and thus most Table Maps contain a single ttlimits symbol record.

These are text format strings with the same structure as the ttlimits records described above. They are intended to be used to extract blind/limit information from the colimits regions using text transforms. As a poker site could certainly paint the various blind/limit information in different locations in the window, it is very likely that multiple colimits records are required, if any are needed at all.

This record describes the name of the poker site that this Table Map is for. It is good practice to include the number of chairs and game flavor that this Table Map is intended to be used for. You might have a "fulltilt9sng", "fulltilt6ring" and

c0limits, c0limits0-c0limits9

sitename

"fulltilt2ring" for different Table Map files for Full Tilt's various table layouts and flavors. This record is what is queried via the OpenHoldem "sitename\$" symbol.

The name of the poker network that this Table Map is for. This is used to determine the appropriate network to query Poker Tracker data for when s\$sitename is not a Poker Tracker-supported site. OpenHoldem will first check s\$sitename for a supported network name, then s\$network.

Both s\$sitename and s\$network are not casesensitive. OpenHoldem converts them internally to lowercases.

The following names are supported for PT2:

- stars
- party
- ultimate
- absolute
- microgaming
- ongame
- cryptologic
- pacific
- fulltilt
- b2b
- tribeca
- worldpex
- ipoker
- tain,
- bodog
- everest
- boss
- betfair

The following names are supported for PT3:

- stars
- party

network

- fulltilt
- ipoker
- everest
- ongame
- boss
- cereus
- pacific
- b2b
- microgaming
- cake
- bodog

The number of chairs that this Table Map supports. The value of this record is what is returned via the OpenHoldem "nchairs" symbol.

Used by the Autoplayer what buttonßsequence to click to go allin 0: click only "max" (or "allin") non-zero: first click "max" (or "allin"), then click "raise" as confirmation

Used by the Autoplayer to determine how to enter the f\$betsize amount into the betsize entry box on the casino window. This is necessary, as bets and raises get interpreted differently by different casinos.

- 1. Enter (f\$betsize currentbet call)
- 2. Enter (f\$betsize currentbet)
- 3. Enter f\$betsize without any adjustment.

Default: 3

As an example let us assume, that you are short-stacked in a SNG. Your balance is 9 big blinds before you have to post the big blind. Everybody folds and button raises to 3 big blinds. You want to push (f\$betsize = 9). What amount should you enter for different casinos?

nchairs

allinconfirmationmethod

betsize interpretation method

- 1. 6 bbs the amount that exceeds the previous raise of 3 bbs.
- 2. 8 bbs the amount you have to add to your current bet of 1 bb.
- 3. 9 bbs your future current bet after have you raised.

This is used to determine the appropriate site interpretation for the contents of c0pot0:

- 1. Common pot
- 2. Total pot, includes current bets and side bets
- 3. Largest of all pots (common pot and sidepots)

Default: 1

These symbols control which OCR algorithm will get used for the specific text transform 0-3. For example if the symbol h3type is defined as fuzzy then all regions with transform Text3 will use the fuzzy algorithm. The default value is "plain" if no symbol exists for a given group. The possible values for these symbols are "fuzzy", "plain", or a specific tolerance value. The default fuzzy text tolerance is 25%, but can be overridden by stating the value in this symbol, such as "0.355", which would cause the scraper engine to use the fuzzy text algorithm with a 35.5% tolerance value. Values of 0 are the same as "plain". New Symbol in version 2.0; previously, this was a global preference set from the OpenHoldem preferences GUI. This symbol instructs the Autoplayer how to

take the first action in a betsize sequence. A

potmethod

t0type - t3type

betsize sequence always consists of:

- 1. select the text in the betsize box
- 2. delete the text in the betsize box
- 3. enter the new bet in the betsize box
- 4. confirm the betsize bet amount.

Valid values for this Symbol are available in a drop down control and are:

- "Sgl Click" Single left click on the betsize box to select the existing text
- "Dbl Click" Double left click on the betsize box to select the existing text
- Triple Click' Triple left click on the betsize box to select the existing text
- "Click Drag" Left click on the leftmost pixel of the betsize box, drag to the rightmost pixel of the betsize box, and then release the left mouse button to select the existing text
- "Nothing" Skip the text selection step of the betsize action

Default: None, must be explicitly specified New Symbol in version 2.0; previously, this was a global preference set from the OpenHoldem preferences GUI. This symbol instructs the Autoplayer how to take the second action in a betsize sequence. A betsize sequence always consists of:

- 1. select the text in the betsize box
- 2. delete the text in the betsize box
- 3. enter the new bet in the betsize box
- 4. confirm the betsize bet amount.

Valid values for this symbol are available in a drop down control and are:

betsizeselectionmethod

betsizedeletionmethod

- "Delete" Press the "Delete" key to delete the selected text in the betsize box
- "Backspace" Press the "Backspace" key to delete the selected text in the betsize box
- "Nothing" Skip the text deletion step of the betsize action

Default: None, must be explicitly specified

New Symbol in version 2.0; previously, this was a global preference set from the OpenHoldem preferences GUI. This symbol instructs the Autoplayer how to take the fourth action in a betsize sequence. A betsize sequence always consists of:

- 1. select the text in the betsize box
- 2. delete the text in the betsize box
- 3. enter the new bet in the betsize box
- 4. confirm the betsize bet amount.

Valid values for this Symbol are available in a drop down control and are:

- "Enter" Press the "Enter" key to confirm the betsize bet entry
- "Click Bet" Click the "Bet" button to confirm the betsize bet entry
- "Nothing" Skip the bet confirmation step of the betsize action

Default: None, must be explicitly specified

New Symbol in version 2.0; previously, this was a global preference set from the OpenHoldem preferences GUI. This symbol instructs the Autoplayer how to click any

betsizeconfirmationmethod

buttonclickmethod

button defined by the iXbutton region records.

Valid values for this Symbol are available in a drop down control and are:

- "Single" Single left click on buttons
- "Double" Double left click on buttons

Default: "Single"

Some casinos require additional input to confirm betpot actions. Valid options:

betpotmethod

- None
- Raise click the raise-button after having clicked a betpot button.

use_comma_instead_of_dot

Some casinos require the use of the "comma" as a decimal separator rather than the "dot". This setting will instruct OpenHoldems autoplayer to use a comma if needed.

handnumber_min_expected_digits

Optional parameter to specify the minimum number of digits for a valid handnumber to reduce the risk of bad scrapes and incorrect handresets.

handnumber_min_expected_digits

Optional parameter to specify the maximum number of digits for a valid handnumber to reduce the risk of bad scrapes and incorrect handresets.

- Default: scraping first cardbacks, then nocard, then cardfaces (or ranks + suits). Works for most casinos, optimal performance and also serves the case where cards and cardbacks are at different locations
- "1": scraping first card-faces, then cardbacks, finally nocard. This method

cardscrapemethod

is necessary at casinos that display additional cardbacks also for players that hold face-cards.

chipscrapemethod

islobby

This Symbol instructs the "chip stack counter" within the screen scraper engine to act in one of three distinct modes of operation: If this Symbol is not present, or does not match one of the other formats, the "chip stack counter" begins counting chips at the 00 Region record, stops counting a given stack when a chip match is not found, and stops counting all stacks when the first chip in a given stack is not matched. This is the default, pre-2.0 behavior. If this symbol is set to "All", then the scraper engine will check all 200 possible chip locations (10 stacks of 20 chips) every time a chip count is required. If this symbol is set to the format "XxY", where X is an integer between 1 and 10 representing the number of stacks, and Y is an integer between 1 and 20 representing the number of chips in each stack, then the scraper engine will check X times Y chip locations every time a chip count is required. If this symbol exists and is non-zero OpenHoldem will know that it is connected to a lobby window (for table.positioning and shutdown-handling). The tablemapcompleteness checker will be turned off and the OH-script-symbol islobby will become true. Lobby-windows will always be positioned at the top-left of the screen where absolute coordinates equal relative windowcoordinates to make hoppers more robust in case of lost focus.

If this symbol exists and is non-zero OpenHoldem will know that it is connected ispopup

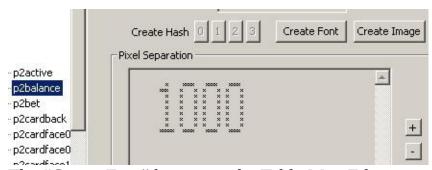
to a popup window. The tablemapcompleteness checker and the tablepositioner will be be turned off and the OHscript-symbol ispopup will become true. This feature can be used to connect OpenHoldem to known popup-windows with a minimalistic tablemap and then click confirmation-buttons with the i86x-regions or do other simple hopper-tasks.

Font Records

Description

Font records are used as lookup records for Regions that utilize Text transformations. When a Text transformation is required for a region, the foreground pixels in that region will be scanned left to right, and the widest font record that matches those pixels will be considered a match. The region will continue to be scanned left to right until all the pixels in the region are consumed. The return value of that region's text transform is this entire sequence of ASCII characters.

Manipulating font records using OpenScrape



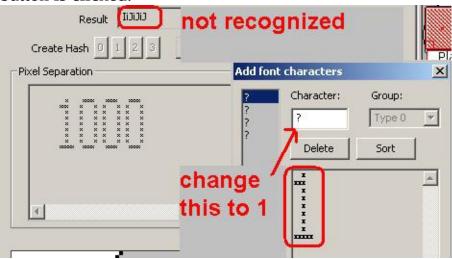
The "Create Font" button on the Table Map Editor window is used to create new font records. This button is only available if a Text transform is specified for the selected region record.

To create new font records:

- 1. Select a Region record that has a text transform
- 2. Ensure that the Region's color parameters provide good separation of foreground and background pixels (preview in the Pixel Separation field)
- 3. Click the "Create Font" button

The number of the text transform in the region (Text0, Text1, Text2, or Text3) determines which Font record group (0 to 3) the new fonts will be added to. Multiple font record groups may be necessary if the casino uses multiple fonts or if fonts are blinking or otherwise highlighted when it is a players turn. Assuming that some of the pixels in the selected region represent unknown font characters, then the following window will appear when the "Create Font"

button is clicked:



On this window, a list of the individual font characters that OpenScrape thinks it has found is displayed in the list on the left side. As we have not told OpenScrape what character each of these pixel groups represent, they are displayed as question marks. (Note that the question mark can be considered a valid font character as well, so if you use this character in your Table Map, be aware of this fact).

For each unknown character in the list on the left, click to select it, then either: Enter which character is represents in the "Character" field (in this example, it is evident that the grouping of foreground pixels represents the capital letter P), or Click on the "Delete" button if this is not a character you want stored in the Table Map for some reason, such as it is a duplicate, or the automatic character separation didn't work quite correctly

Clicking on the "Sort" button will arrange the characters alphabetically in the list, which might make them easier to process.

When you dismiss this window by clicking on the "OK" button, OpenScrape will add each of the characters in the list on the left, their ASCII representation, and their pixel layout to the list of Table Map records. This includes duplicates, and unknowns (question marks), so be sure to remove these prior to clicking "OK", or your Table Map might get filled with superfluous Font records.

Fuzzy fonts

"Fuzzy fonts" are sometimes used by casinos when displaying text on their poker interfaces. You can identify if a casino uses fuzzy fonts by looking closely at the pixel patterns for individual characters as they are displayed at different times or in different locations on the screen. (Hint: use zoom) As an example, if

when you look at the capital "A" character as it is displayed for seat 0's name vs. seat 1's name, you can see that those two letters use different pixels, of different colors, especially on the slanted arms of the "A", you generally will need to use fuzzy font recognition.

The general procedure for defining fuzzy font records is as follows:

- 1. Decide which fonts will be fuzzy, and create those in a different Font group from your other fonts (we will use group 1 "Text1" in this example). Fonts can be created with the font color defined as the foreground pixel color, or the background color as the foreground pixel color. In the case of the latter, the so-called inverse fonting may sometimes give better results. Experiment! If using the font color for the foreground pixel color, it is probably best to make the fonts as thick as possible using a large RGB color cube radius. If using the background color as the foreground pixel color, a radius of zero probably works best.
- 2. Create a t1type Symbol record (for the Text1 group), and set the free-form text to "fuzzy" for default tolerance (25%), or to a specific tolerance value (0.50 for 50%, 0.33 for 33%, for example). Experiment to find the best tolerance all casinos are different in how they display fuzzy fonts. If this symbol is set to anything besides "fuzzy" or a floating point value, then the engine will used non-fuzzy font recognition engine.
- 3. You can also create multiple samples for various characters, and it will help. The numbers "6", "8", and "3", for example, are similar enough that distinguishing fuzzy versions of these is difficult. Creating more font records (samples) for the various 6's, 8s and 3's will help with the recognition.

Technical Reference

In the Table Map (.tm) file, these records will be preceded with the characters "t\$". Font records have the following format:

$$T < n > $ < a > < x < 0 > < x < 1 > < x < 2 > ... < x < 3 > >$$

<n> is the font group (0-3) that this font record belongs to <a> is an actual printable (non-whitespace) case-sensitive ASCII character (note that the angle brackets <> are not used) <x0> to <x30> are 32bit hexadecimal values that describe the foreground pixels for the character. The maximum individual character width is 31 pixels. These values describe the on-off state of the pixels for the character, with 1 meaning on (foreground) and 0 meaning off (background).

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T16:02:16.359000

Hash Point Records

Description

The purpose of Hash Point records is to define the pixels within a region that are used in a hash calculation. The "Hash0" transform always uses all pixels in a region to calculate its hash, so these records are only relevant for regions that use "Hash1", "Hash2", or "Hash3" transforms.

There is a one to one correlation between the "Hash1", "Hash2", or "Hash3" transforms and the group 1, 2 and 3 Hash Point records. For example, a region that calls for a "Hash2" transform will use the pixels specified by the group 2 Hash Point records in the hash calculation.

The points defined for a Hash Point group must fall within the boundaries of the regions that you are calculating a hash for. Hash Points that fall outside of the region will be ignored, as they are invalid. For example, if you have card regions that are 10x20 pixels, then hash points (11,15) and (8,22) will be ignored, whereas hash points (5,5) and (9, 18) will be used in the hash calculation.

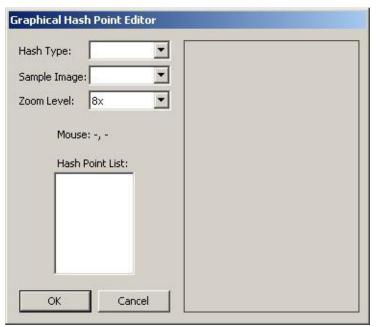
Manipulating hash point records using OpenScrape

There are two ways to create new Hash Point records:

1.Select the "Hash Points" category in the record tree display on the Table Map Editor window, and click the "New" button. A window will be displayed that allows you to enter the hash point group and the X and Y parameters:

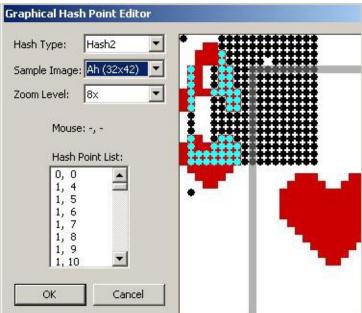


2.Somewhat less intuitively, but ultimately easier to use, is to select the "Hash Points" category in the record tree display on the Table Map Editor window, and click the "Edit" button. A window will be displayed that allows you to select the hash point group, a sample image, and then interactively define the Hash Points for that group:



When you select a Hash Point group from the "Hash Type" control, the "Hash Point List" box will populate with all the Hash Points currently defined for that group. In this case, the box is empty, as we have not yet defined any Hash Point records for group 1.

After selecting an Image record in the "Sample Image" control (note you need to have Image records created first for this to work), the right side of the window will display that image and overlay the Hash Points for the selected group, like so:



The zoom level of the sample image can be adjusted with the "Zoom Level" control. Adding Hash Points can be done interactively by left-clicking on the

sample image where the point is needed. Right clicking on an existing point in the sample image will remove it from the group.

Clicking the "OK" button will then make the changes (adds and deletes) to the Table Map Hash Pointe record list. To edit an existing Hash Point record, select the record in the record tree display on the Table Map Editor window, and click the "Edit" button. A window will be displayed that allows you to edit the hash point group and the X and Y parameters. To delete an existing symbol record, select the record in the record tree display on the Table Map Editor window, and click the "Delete" button. A window will be displayed asking for verification of this action. An entire group of Hash Points can also be edited and/or deleted by selecting the "Hash Points" category in the record tree display on the Table Map Editor window, and clicking the "Edit" button. This will bring up the Graphical Hash Point Editor window, as described above.

Note: The Insert and Delete keys can also be used as a shortcut for adding and deleting individual Hash Point records.

Technical Reference

In the Table Map (.tm) file, these records will be preceded with the characters "p\$". Hash point records have the following format:

$$p < n > $ <_X > <_y >$$

<n> is the group (1-3) that this hash point record belongs to <x> and <y> are the coordinates of the point, relative to the region's rectangle, the top left corner of which is 0.0.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T20:55:11.609000

Hash Records

Description

The purpose of Hash Records is to allow for the rapid lookup of pixel patterns, without having to scan through each pixel in a region to find a match. For example, if we have a card region specified to use a "Hash0" transform, the screen scraper engine will first reduce the pixels in that region to a 32-bit hash value. (A link to Bob Jenkins' hash algorithm that is used in this project can be found in the references section) Once the screen scraper engine has calculated the 32-bit hash value, it looks to see if there exists a matching 32-bit value in the associated group of Hash Records, in this example, group 0. If a matching 32-bit hash value is found, the screen scraper engine returns the <name> of the Hash Record to the game state engine.

Manipulating hash records using OpenScrape

Hash Records are created from specific Image records, so in order to create a Hash record, you must first have collected the associated Image record (see section {4.9}).



The "Create Hash" buttons on the Table Map Editor window are used to create new Hash records. Button "0" creates Hash records in the Hash0 group, button "1" creates Hash records in the Hash1 group, and so on. These buttons are only available if an Image record has been selected in the tree display on the Table Map Editor window.

Note: Only one Hash record can exist in a Table Map in each hash group for any given Image record. Otherwise, there would be collisions, and the screen scraper engine would not know which Hash record was the correct match. In the "Create Hash Record buttons" screenshot above, Hash records for group 0 and 2 have already been created for the selected Image record, so those buttons are grayed out.

To create a new Hash records:

- 1. Select an Image record
- 2. Click the "Create Hash" button for the hash group that you want to create

this Hash record in.

The following window will then appear to confirm the addition of the new Hash record:



When you add a new Hash record with the "Create Hash" buttons, OpenScrape will calculate the hash value for you and add it to the record. (This is new behavior in 2.0; previous versions stored a zero hash value on new record creation) However, sometimes you may need to change that stored hash value, if for example, you replace the underlying Image record, or you change the number or location of the related Hash Points. If one or more hash values need to be updated, use the "Update Hashes" option on the Edit manu on the Main window, and the hash values on all Hash records will be re-calculated. Technical Reference In the Table Map (.tm) file, these records will be preceded with the characters "h\$". Hash records have the following format:

h<n>\$<name> <value>

<n> is the group (1-3) that this hash record belongs to <name> is the string that is returned to the scraper engine upon a match (the "result") <value> is a 32bit hexadecimal value that contains the calculated hash.

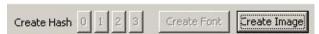
Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T20:57:18.750000

Image Records

Description

Image records serve multiple purposes. The first purpose is for regions that use the Image transform. When a region requires this type of lookup, the screen scraper engine will first look for an exact pixel-for-pixel image match in the Image records, and if no match is found, will then look for the closest match (with the 65% tolerance, as described in the Region Records {Technical Reference} section). The second purpose of the Image records is to serve as a reference from which Hash records are created (see the {Hash Records} section). Note: Image transforms are very CPU intensive, as they require 2 passes through the Image records, the first to try to find an exact match, and the second to find a closest match. Each pass requires a comparison of every single pixel in the region. For large regions, this is a lot of comparisons. The time required to do Image transform lookups is directly proportional to the number of Image records present, and the size of the requesting Region record. It almost always makes more sense to use the Hash transform to do image matching, as the Hash lookup algorithm is orders of magnitude faster.

Manipulating image records using OpenScrape



The "Create Image" button on the Table Map Editor window is used to create new Image records. This button is only available if a region record has previously been selected.

When the "Create Image" button is clicked, the following window will appear:



The only required piece of information on this window is the name of the Image record, however, the name you choose is crucially important, as this name is what is returned to the game state engine directly for an Image transform, and is what is populated on the Hash record when you create a hash from this image. The name is dependent on what the image is used for, but for example, if this is

an image of the ace of clubs, you would want to name it "Ac". If this is an image of a dealer button, you would want to name it "dealer". (See the Region Records section, {Technical Reference}, for valid return values for each region)

Technical Reference

In the Table Map (.tm) file, these records will be preceded with the characters "i\$". Image records have the following format:

i\$<name> width height scanline0

scanline1

...

scanlineN

<name> is the description of the image, and is the value returned to the screen scraper engine or is the name given to an associated Hash record <width> and <height> are integer values that give the size of the image <scanline0> - <scanlineN> contain the raw 32bit hexadecimal values for each pixel. Each pixel is in the form BBGGRRAA. There is no whitespace between pixel values. Each scanline is terminated with a standard Microsoft Windows line ending (carriage return/line feed). The number of scanline records is equal to the height of the image. The text length of a scanline record is width*8.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-18T21:07:21.734000

The Bot-Logic

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:16:58.562000

Built-in Symbols

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-23T18:00:30.671000

Built-in Symbols

This section is largely based on the information originally written by Ray Bornert (http://www.winholdem.net/help/help-formula.html#symbols) and extended to include OpenHoldem-specific symbols.

These symbols take output from the scraper engine, from Table Map parameters, and from your formula functions to derive various bits of data that are useful when instructing the OpenHoldem Autoplayer how to act. The set of symbols got extended over time — not all of them are available for all versions of OpenHoldem. This list reflects the state of the latest version. Some old WinHoldem symbols got never implemented due to lack of documentation and practical relevance. Others got replaced or removed over time. Please refer to the chapter "Outdated Features and Incompatibilities between OpenHoldem and WinHoldem" for more information.

As you already might have noticed some of these symbols are of rather technical nature, A set of several hundred more poker-logical symbols is included in the OpenPPL-library that gets discussed in the seeparate OpenPPL-manual. Both languages can be mixed in nearly any way however, so please read on. Here you will find some valuable symbols that you might be miss in OpenPPL.

General

Symbol Meaning

session the current logging instance (0-24)

version returns the version number of OpenHoldem that is

currently running

Casino and Poker Table

Symbol Meaning

Window handle of the poker table

attached_hwnd Provided for DLL- and Perl-devlopers who can

now directly access the poker-table if needed

nchairs the integer value for the Table Map symbol

s\$nchairs

true if user defined string "def" appears within the

network\$def Table Map symbol s\$network; case-sensitive;

lower-cases are expected

true if user defined string "abc" appears within the

sitename\$abc Table Map symbol s\$sitename; case-sensitive;

lower-cases are expected

true if user defined string "abc" appears within the

title of the attached poker-table; case-sensitive;

islobby true if the tablemap is designed to connect to the

lobby

ispopup true if the tablemap is designed to connect to

known popup windows

Limits

istournament

Symbol Meaning

bblind the big blind amount sblind the small blind amount

ante the current pre-deal ante requirement

buyin the tournament buyin in dollars

lim the current table limit 0=NL 1=PL 2=FL

isnl $(\lim == 0)$ No Limit ispl $(\lim == 1)$ Pot Limi isfl $(\lim == 2)$ Fixed Limit

true if you play a double-or-nothing tournament.

isdon The conditions for istournament must apply and the

title must also contain a DON identifier

true if you play a multi-table-tournament.

ismtt The conditions for istournament must apply and the

title must also contain a MTT identifier

true if you play a single-table-tournament

issng All tournaments that can't be recognized neither as

MTT nor as DON will qualify as SNGs..

true if any type of tournament table is detected. This feature works automatically, based on title-

strings of the poker-table, blind-levels, sum of

chips at the table, etc.

isfinaltable true if you play the final table of a MTT, depending

on the tablemap region r\$c0isfinaltable

true if OpenHoldem is connected to a lobby

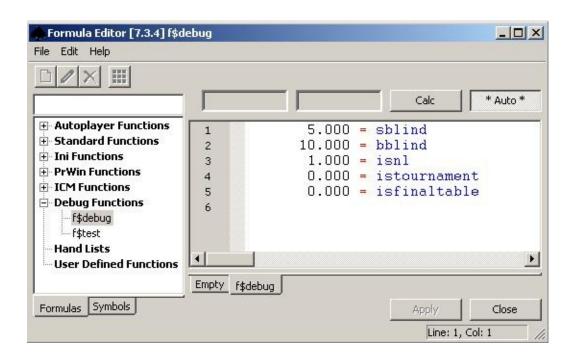
islobby window (for window placement and shutdown).

Depends on the TM-symbol s\$islobby

true if the game is a rush / zoom style cash-game

isrush where you get moved to a new table immediately

when you fold.



Tournament Info

MTT symbols provide you some info about the tournament you currently play. They are most useful forswcisions at the bubble. Each one of these symbols returns the value of the corresponding tablemap-region (e.g. r\$mtt_number_entrants, etc.). To make use of these symbols they must be displayed at the table, so that OpenHoldem can scrape them.

Symbol Meaning

mtt_paid_places Number of paid places, useful for decisions at the

bubble

My current place in the tournament depending on

mtt_my_rank stacksize, assuming that all players with smaller

stacks bust before me

mtt_largest_stack k Stacksize of the chip-leader, usually in dollars

mtt_average_stack

Average stacksize of all tournament plazers,

usually in dollars

mtt_smallest_stack

Smallest stack in the tournament, usually in

dollars

Hand Rank

Handrank rates your hand-value, assuming you go preflop allin against N opponents, who hold random cards (N = f\$prwin_number_of_opponents). A low handrank means. your hand is good, a high handrank means, that many holdings are a favourit against you. For more info see the chapter about handrank.

Symbol Meaning

handrank169 your pocket holdem hand rank 1-169 handrank2652 your pocket holdem hand rank 12-2652 your pocket holdem hand rank 6-1326

(handrank2652/2)

handrank1000 your pocket holdem hand rank 4-1000

(1000*handrank2652/2652)

handrankp 2652 / (1+f\$prwin_number_of_opponents)

Chairs

Please note OpenHoldem needs to see player-cards in a non-showdown-situation to recognize the userchair and then will lock the chair for stability reasons until the session of the hand. A change in userchair can happen on table-change in MTTs at some casinos and will be handled automatically.

Symbol Meaning

userchair user chair number (0-9), 0 is usually top right

headsupchair chair number of the one and only opponent headsup (0-9)

bigblindchair big blind chair number(0 - 9) smallblindchair small blind chair number (0-9)

dealerchair dealer chair number (0-9) cutoffchair cut off chair number (0-9)

mp3chai middle position 3 chair number (0-9) mp2chai middle position 2 chair number (0-9) mp1chai middle position 1 chair number (0-9) ep3chair early position 3 chair number (0-9) ep2chair early position 2 chair number (0-9) ep1chair early position 1 chair number (0-9) utgchair under the gun chair number (0-9)

firstcallerchair the chair of the first caller in front of you (0-9), current orbit

only

lastcallerchair the chair of the last caller in front of you (0-9), current orbit

only

firstraiserchair the chair of the first raiser in front of you (0-9), current orbit

only

lastraiserchair the chair of the last raiser in front of you (0-9), current orbit

only

raischair synonym for lastraiserchair

bigstackchair the chair of the playing opponent with the biggest stack (0-9) smallstackchair the chair of the playing opponent with the smallest stack (0-9)

chair\$abc player "abc" chair number (0-9); -1 if not found

chairbit\$abc player "abc" chairbit (1 << chair\$abc); 0 if not found

Positions

To avoid problems with fast folds in early position the dealposition-symbols will consider the following players as dealt

- any players who are betting, especially the blinds
- all active players between the blinds and the first user who holds cards, i.e. players who are seated and not sitting out. If one of these players does not hold cards, then OpenHoldem will assume that it is due to very fast folds in early positions.
- the user, if he is seated and holds cards
- any players with cardbacks, especially all players between the user and the dealer.

Symbol	Meaning
betround	betting round (1-4) 1=preflop, 2=flop, 3=turn, 4=river
currentround	synonym for betround
previousround	the previous betting round (1-3, 1 if the current round is preflop)
preflop	verbose constant for the 1st betting round
flop	verbose constant for the 2nd betting round
turn	verbose constant for the 3rd betting round
ruver	verbose constant for the 4th betting round
betposition	your bet position (1=left to the dealer=usually sblind, 2=bblind,, nplayersdealt=dealer); contrary to dealposition betposition will change as

dealposition

callposition

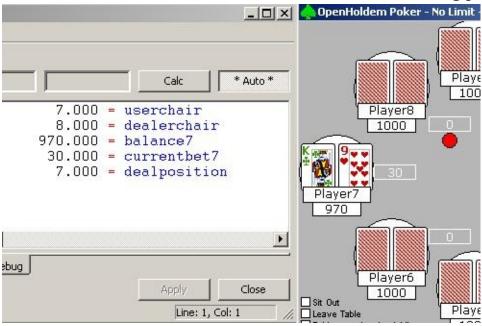
dealpositionrais

betpositionrais

players fold in front of you. your deal position 1 = left to the dealer, usually small blind2 = 2 left to the dealer, usually big blind ... nplayersdealt = dealer. dealposition will not change as players fold. Dealposition does not magically auto-adapt for special cases like a missing small-blind. If you want to figure out chairs better use the more modern chair symbols.

your numbered offset from the raising player (who is 0)

the deal position of the raising player (1-10) the bet position of the raising player (1-10)



Probabilities

There are several variations of PrWin (standard, weighted and enhanced), each with several pros and cons. For more info see the chapter about PrWin.

Symbol Meaning

the probability of winning this hand (0.000 - 1.000)

•

PrWin might assume:

• concrete ranges for your opponents with weightings for each possible holdings if you use enhanced PrWin at the DLL-level.

• simplified, but reasonable ranges if you use weighted PrWin at the formula-level, making use of f\$prwin_mustplay, f\$prwin_top_clip, f\$prwin_willplay and f\$prwin_wontplay

random ranges of your opponents (default),
 where f\$prwin_number_of_opponents
 specifies the number of opponents for the
 prwin-simulation to control the over-optimism
 of this very simple model.

the probability of losing this hand (0.000 - 1.000),

assuming random ranges of your opponents

the probability of pushing this hand (0.000 - 1.000)

, assuming random ranges of your opponents

probability that all opponents have a lower hand

right now, assuming random ranges of your

opponents

probability that any opponents have a higher hand right now random, assuming random ranges of your

opponents

number between (0.000-1.000). Value is

recalculated each time symbol appears in formula. random number between (0.000-1.000) for the curren heartbeat. Value is calculated only once in

prwin

prlos

prtie

prwinnow

prlosnow

random

randomheartbeat current heartbeat.

Introduced for OpenHoldem 2.1.9

random number between (0.000-1.000) for the randomround

current round. Value is calculated only once in

current round.

random number between (0.000-1.000) for the randomhand

hand. Value is calculated only once per hand.

Formulas

Symbol Meaning

reference the specified standard or user defined f\$name

function

Chip Amounts

Symbol Meaning

balance your balance

balanceatstartofsession your first known balance after OpenHoldem

connects to a table

balance0..balance9 specific player/chair balance

stack0..stack9 sorted playersplaying balances from 0=biggest to

9=smallest

ranked list of player balances similar to

balance_rankx (x=0-9) stack0..stack9, but includes players not currently in

hand and includes currentbet for each player as well. rank0 has highest balance, rank9 smallest

currentbet your current amount of chips in play

currentbet0..currentbet9 specific player/chair currentbet call the amount you

need to call

bet the amount of a single initial bet or raise for current

round

bet1..bet4 the amount of a single initial bet or raise for round

1 - round 4

the total amount of chips in play including player

bets

potcommon the total amount of chips in the middle

potplayer the total amount of chips in front of all players

Number of Bets

Please note that in a fixed-limit game (default) the bets double up at the turn. This will affect the symbols below. If you play no-limit or pot-limit please make sure that your tablemap detects the game-type correctly.

Symbol Meaning

nbetstocall total number of additional bets required to call.

total number of additional bets required to min

nbetstorais raise.(Fixed Limit)

ncurrentbets total number of bets currently in front of you.

ncallbets total number of bets you would have on the table if

you call

nraisbets total number of bets you would have on the table if

you min-raise (Fixed Limit)

List Tests

Symbol Meaning

true if your hand is in named list XYZ

Please note that OpenHoldem 5.x+ supports

listXYZ arbitrary many lists with user-defined names as

long as the name starts with "list". OpenHoldem is

no longer restricted to list0..list999.

Poker Values

Symbol Meaning

pokerval absolute poker value for your 5 card hand

pokervalplayer absolute poker value for your 2 card pocket hand

only

pokervalcommon absolute poker value for the common cards

bit list of where your pocket cards are used in your

5 card hand.

If a certain hand can be made with either one of

your cards or with one of the board cards

OpenHoldem will first take your card. This can happen only for straights and for kickers for quads. This behaviour may seem counter-intuitive to some

people, but got inherited from WinHoldem and kept

for backward-compatibility.

• Your hand: JTs

• Board: 789T

• Here you have a straight using "both" of your cards in the top two places. So pcbits will be

0b11000 = 24 here.

number (0-2) of your pocket cards used in your 5

card hand.

As npcbits symbol depends on pcbits, the same

word of caution applies to this symbol, too.

pcbits

npcbits

Poker Value Constants

```
Symbol
            Meaning
            1<< 0 (2 ** 0)
hicard
            1<<24 (2 ** 24)
onepair
            1<<25 (2 ** 25)
twopair
threeofakind 1<<26 (2 ** 26)
            1<<27 (2 ** 27)
straight
            1<<28 (2 ** 28)
flush
            1<<29 (2 ** 29)
fullhouse
fourofakind 1<<30 (2 ** 30)
straightflush 1<<31 (2 ** 31)
royalflush
            0x800edcba
```

Hand Tests

Symbol Meaning

\$CCc references a hand symbol - see the *Hand*

Symbols section below for details

\$\$CCc references a card symbol - see the *Card*

Symbols section below for details

ishandup true if your hand has gone up a level (i.e. from

1 pair to 2 pair)

ishandupcommon true if common hand has gone up a level (i.e.

from 1 pair to 2 pair)

ishicard true when you have hicard hand isonepair true when you have one pair istwopair true when you have two pair

isthreeofakind true when you have three of a kind

isstraight true when you have a straight isflush true when you have a flush

isfullhouse true when you have a full house isfourofakind true when you have four of a kind isstraightflush true when you have a straight flush isroyalflush true when you have a royal flush

OpenPPL symbol that ranks the strength of

nutfullhouseorfourofakind quads and full houses, 1 = best, 999 = not at

all

nutfullhouseorfourofakind_ntotal total number of currently possible quads and

full houses, for comparison

Hand and Board-Expressions

hand\$XYZ

true, if your hand contains XYZ. The expression may

contain 1 or 2 cards, optionally concrete suits or

optionally the postfix "Suited".

true, The expression may contain 1 up to 5 cards,

board\$XYZ optionally concrete suits or optionally the postfix

"Suited".

Hand and board expressions got implemented for OpenPPL, but of course they can be also used for OH-script. Especially board-expressions are in most cases more intuitive than rankbits(common). Some examples for hand and board-expressions (OpenPPL)

WHEN hand\$AA RaiseMax FORCE
WHEN hand\$AKSuited RaisePot FORCE
// We love playing hearts...
WHEN hand\$Ah Call FORCE
// ...but fear the devil
WHEN board\$666 Fold FORCE

Pocket Tests

Symbol Meaning

true when your two dealt pocket cards are rank ispair

equal (0-1)

true when your two dealt pocket cards are suit issuited

equal (0-1)

true when your two dealt pocket cards are rank isconnector

adjacent (0-1)

Pocket / Common Tests

Symbol Meaning

true when you have a pair with the highest boardishipair

card or an overpaor

true when you have a pair with the lowest boardislopair

card or an underpair

true when you have a pair that is neither hipair nor ismidpair

lopair

true when you have the highest straight possible ishistraight ishiflush true when you have the highest flush possible

Players, Opponents

Note that "raising players" does no longer count the so-called "blind-raisers" (former WinHoldem definition).

Symbol Meaning

maximum number of dealt players seen in this

session.

Especially meant to distinguish certain gametypes maxnplayersdealt (FR / SH / HU) in tournaments. The alternative

(FR / SH / HU) in tournaments. The alternative symbol nchairs is only partially suitable as it depends only on the tablemap shich might fit to

multiple gametypes.

nplayersseated number of players seated (including you) (0-10)
nplayersactive number of players active (including you) (0-10)
nplayersdealt number of players dealt (including you) (0-10)
nplayersplaying number of players playing (including you) (0-10)

nfriendsseated 1 if you are seated, 0 otherwise (0-1)

number of opponents seated (not including you) (0-

9)

nopponentsactive number of opponents active (not including you) (0-

9)

number of opponents dealt (not including you) (0-

9)

nopponentsplaying number of opponents playing (not including you)

(0-9)

number of opponents playing with a zero current

bet equal to the previous bettor (0-9)

number of opponents playing with a non-zero

current bet equal to the previous bettor (0-9)

number of opponents playing with a current bet greater than the previous bettor (0-9). This symbol counts only people who are truely raising, e,g, more

nopponentstruelyraising than a big-blind preflop. It does not count people

posting antes or blinds like the former WinHoldem symbol "nopponentsraising", Also counts the first

better postflop.

number of opponents playing with a non zero

current bet (0-9)

number of opponents that have folded this hand (0-

9)

nplayerscallshort number of players that must call to stay in the hand

nchairsdealtleft number of chairs dealt after your chair, up to and

including the dealer, but not the blinds.

nchairsdealtright number of chairs dealt before your chair, including

the blinds

playersseatedbits bits 9-0: 1=seated 0=unseated playersactivebits bits 9-0: 1=active 0=inactive playersdealtbits bits 9-0: 1=dealt 0=notdealt

playersplayingbits bits 9-0: 1=playing 0=notplaying opponentsseatedbits bits 9-0: 1=seated 0=unseated opponentsactivebits bits 9-0: 1=active 0=inactive opponentsdealtbits bits 9-0: 1=dealt 0=notdealt

opponentsplayingbits bits 9-0: 1=playing 0=notplaying

Flags

Symbol	Meaning
fmax	highest numbered flag button pressed
f0f19	true if flag 0 - flag 19 button is pressed, false otherwise
flagbits	flag button bits 19-0 - 1=pressed 0=notpressed
0 1 2 3 4 5 6	7 8 9 10 11 12 13 14 15 16 17 18 19

The flags toolbar is used to set internal OpenHoldem flag symbols f0..f19 to on or off. These flags can be used in your formula set to quickly enamble or disable certain functionality. E.g. you could use flag 0 to to tell your bot to use a different starting-hands-chart or to do more conti-betting.

##f\$preflop##

WHEN fO RETURN f\$PokerTragedySHCAction FORCE WHEN Others RETURN f\$SklanskySHCAction FORCE

Common Cards

Symbol Meaning

number of common cards known (normal not

highlighted)

(Un)known Cards

Symbol	Meaning
ncardsknown	total number of cards you can see (yours and commons)
ncardsunknown	total number of cards you cannot see (deck and opponents)
ncardsbetter	total number of single unknown cards that can beat you, e.g. if the board is four suited in hearts, and you have two spades, then ncardsbetter will be at least 9, because of the possible flush
\strikeout off\uuline off\uwave offnouts	\strikeout off\uuline off\uwave offthe total number of unseen single cards that if dealt to the board might put your hand in the lead. to be counted as an out, the card must be able to bump your level and your new level must be higher than the resulting common level.

Ray E. Bornerts definition of an out is way too optimistic and nearly unusable (a hand like 720 at a board of AKQ will give you 6 outs, as every 7 and every 2 will bump your handlevel to a pair, which beats high-card ace at the board). To calculate outs in a meaningful way, there is f\$mh_nouts in Winngys and HTCs hand-strength-library and there are also some good outs-counters available in the forum. However we had to keep this symbol in the code-base, as some outs-counters are based on nouts and try to discount it in a meanigful way.

nhands

Symbol Meaning

nhands total possible number of two-card hands using the

unseen cards (nhandshi+nhandslo+nhandsti)

nhandshi number of hands that can beat you in a showdown

right now

nhandslo number of hands that you can beat in a showdown

right now

nhandsti number of hands that can tie you in a showdown

right now

Flushes / Straights / Sets

Symbol Meaning

nrankedcommon

nsuited total number of same suited cards you have (1-7)

nsuitedcommon total number of same suited cards in the middle (1-

5)

tsuit specific card suit for nsuited (0-3)

tsuitcommon specific card suit for nsuitedcommon (0-3)

nranked total number of same ranked cards you have (1-4)

total number of same ranked cards in the middle (1-

4)

trank specific card rank for nranked (2-14)

trankcommon specific card rank for nrankedcommon (2-14)
nstraight total number of connected cards you have (1-7)
nstraightcommon total number of connected common cards (1-5)

nstraightfill total number of cards needed to fill a straight (0-5)

nstraightfillcommon total number of cards needed to fill a common

straight (0-5)

nstraightflush total number of suited connected cards you have (1-

7)

nstraightflushcommon total number of suited connected common cards (1-

5)

nstraightflushfill total number of cards needed to fill a straightflush

(0-5)

nstraightflushfillcommon total number of cards needed to fill a common

straightflush (0-5)

Rank Bits

Please note: aces are both hi and lo.

Symbol Meaning

rankbits bit list of card ranks (yours and commons)

rankbitscommon bit list of card ranks (commons) rankbitsplayer bit list of card ranks (yours) rankbitspoker bit list of card ranks (pokerval)

srankbits bit list of suited card ranks (yours and commons

tsuit)

srankbitscommon bit list of suited card ranks (commons

tsuitcommon)

srankbitsplayer bit list of suited card ranks (yours tsuit) srankbitspoker bit list of suited card ranks (pokerval tsuit)

suitbitsplayerN bit list of card ranks(yours) for suit N (N = 0..3),

especially meant for Omaha

suithits common bit list of card ranks (common) for suit N (N = 0..3),

especially meant for Omaha

Rank Hi

Aces are hi.

Symbol Meaning

rankhi highest card rank (14-2) (yours and commons)

rankhicommon highest card rank (14-2) (commons)
rankhiplayer highest card rank (14-2) (yours)
rankhipoker highest card rank (14-2) (pokerval)

srankhi highest suited card rank (14-2) (yours and

commons tsuit)

srankhicommon highest suited card rank (14-2) (commons

tsuitcommon)

srankhiplayer highest suited card rank (14-2) (yours tsuit) srankhipoker highest suited card rank (14-2) (pokerval tsuit)

Rank Lo

Aces are hi.

Symbol Meaning

ranklo lowest card rank (14-2) (yours and commons)

ranklocommon lowest card rank (14-2) (commons) rankloplayer lowest card rank (14-2) (yours) ranklopoker lowest card rank (14-2) (pokerval)

sranklo lowest suited card rank (14-2) (yours and commons

tsuit)

sranklocommon lowest suited card rank (14-2) (commons

tsuitcommon)

srankloplayer lowest suited card rank (14-2) (yours tsuit) sranklopoker lowest suited card rank (14-2) (pokerval tsuit)

Suit Constants

Symbol Meaning

verbose constant to access suit symbols (value = 2),

clubs especially meant for use with the multiplexer-

engine, e.g. "suitbitscommon_clubs"

diamonds verbose constant to access suit symbols (value = 1) verbose constant to access suit symbols (value = 0)

spades verbose constant to access suit symbols (value = 3)

Time

elapsedauto

Symbol Meaning

elapsed time in seconds since sitting down

elapsedhand time in seconds since end of previous hand

time in seconds since autoplayer took

action

elapsedtoday time in seconds since midnight GMT

Autoplayer

issittingin

Symbol Meaning

ismyturn (myturnbits & 7) (rais or call/chec or fold)

bit-vector that tells you what buttons are visible. The bits 43210 correspond to the buttons ARKCF

myturnbits (allin, raise, check, call fold), the legendary

FCKRA in revered order, allin has the highest

value.

True when you are not being dealt out

This symbols is true, if a sit-out-button exists that

can be clicked or if a sit-in button exists that can

not be clicked.

issittingout true when you are being dealt out

isautopost true when you are autoposting

isfinalanswer true when autoplayer preparing to act; false any

other time.

History (I)

The action-symbols (didrais, didcall, etc.) mean: what the autoplayer did, not what the user did or what happened in the hand. That does regularly confuse new users and causes problem when simulating hands at ManualMode or with Replay Frames.

Symbol	Meaning
nplayersround1nplayersround4	number of players that began betting round 1 - round 4
nplayersround	number of players that began the current betting round
prevaction	record of previously attempted autoplayer action. -2 = prefold -1 = fold 0 = check 1 = call 2 = raise 3 = betsize 4 = allin
didfold	the number of times the autoplayer has attempted to fold during the current round
didchec	the number of times the autoplayer has checked during the current round
didcall	the number of times the autoplayer has called during the current round
didrais	the number of times the autoplayer has raised during the current round, i.e clicked the min-raise-button
didbetsize	the number of times the autoplayer has betsized during the current round, i.e. used f\$betsize or one of the betpot-functions
didalli	the number of times the autoplayer has attempted to went allin during the current round

nbetsround1nbetsround4	the largest number of bets in front of any player during round 1- round 4
nbetsround	the largest number of bets in front of any player right now
didfoldround1didfoldround4	true if userchair folded during round 1 - round 4
didchecround1didchecround4	true if userchair checked during round 1 - round 4
didcallround1didcallround4	true if userchair called during round 1 - round 4
didraisround1didraisround4	true if userchair raised during round 1 - round 4
didbetsizeround1didbetsizeround4	true if userchair betsized during round 1 - round 4
didalliround1didalliround4	true if userchair went allin during round 1 -round 4

Versus

Symbol Meaning

vs\$nhands Total possible number of opponent hands

vs\$nhandshi Number of opponent hands that have higher river

chances

vs\$nhandsti Number of opponent hands that have equal river

chances

vs\$nhandslo Number of opponent hands that have lower river

chances

vs\$nhandshinow Number of opponent hands that beat heros hand

currently

vs\$nhandstinow Number of opponent hands that tia gainst hero

currently

vs\$nhandslonow Number of opponent hands that heros hand beats

currently

vs\$prwin Probability (0.000 - 1.000) of winning versus all

possible opponent hands

vs\$prtie Probability (0.000 - 1.000) of chopping versus all

possible opponent hands

vs\$prlos Probability (0.000 - 1.000) of losing versus all

possible opponent hands

vs\$prwinhi Probability (0.000 - 1.000) of winning versus

higher opponent hands

vs\$prtiehi Probability (0.000 - 1.000) of chopping versus

higher opponent hands

vs\$prloshi Probability (0.000 - 1.000) of losing versus higher

opponent hands

vs\$prwinti Probability (0.000 - 1.000) of winning versus equal

opponent hands

vs\$prtieti Probability (0.000 - 1.000) of chopping versus

equal opponent hands

vs\$prlosti Probability (0.000 - 1.000) of losing versus equal

opponent hands

vs\$prwinlo	Probability (0.000 - 1.000) of winning versus lower opponent hands
vs\$prtielo	Probability (0.000 - 1.000) of chopping versus lower opponent hands
vs\$prloslo	Probability (0.000 - 1.000) of losing versus lower opponent hands
vs\$ListX\$prwin	Probability (0.000 - 1.000) of winnning versus hand list X
vs\$listX\$prlos	Probability (0.000 - 1.000) of losing versus hand list \boldsymbol{X}
vs\$listX\$prtie	Probability (0.000 - 1.000) of a tie versus hand list X
vs\$multiplex\$X\$prwin	Probability (0.000 - 1.000) of winnning versus the list specified by function X. The function X (e.g. f\$raisersrange) must return an intereger number, e.g. 0100 and then versus will take the corresponging list (list0list100).
vs\$multiplex\$X\$prlos	Probability $(0.000 - 1.000)$ of chopping versus the list specified by function X.
vs\$multiplex\$X\$prtie	Probability $(0.000 - 1.000)$ of losing versus the list specified by function X.
vs\$prwinhinow	Probability (0.000 - 1.000) of winning versus currently better opponent hands
vs\$prtiehinow	Probability (0.000 - 1.000) of chopping versus currently better opponent hands
vs\$prloshinow	Probability (0.000 - 1.000) of losing versus currently better opponent hands
vs\$prwintinow	Probability (0.000 - 1.000) of winning versus currently equal opponent hands
vs\$prtietinow	Probability (0.000 - 1.000) of chopping versus currently equal opponent hands
vs\$prlostinow	Probability (0.000 - 1.000) of losing versus currently equal opponent hands
vs\$prwinlonow;	Probability (0.000 - 1.000) of winning versus currently lower opponent hands
	Probability (0.000 - 1.000) of chopping versus

vs\$prtielonow currently lower opponent hands

Probability (0.000 - 1.000) of losing versus currently lower opponent hands vs\$prloslonow

History (II)

hi_<sym>x (x=1-4) the value of the symbol <sym> as of your last turn in betting round x. Example: hi_prwin1 would return prwin as of your last turn in br1. Valid values for <sym> are:

PROBABILITIES prwin, prlos, prtie

CHIP AMOUNTS balance..balance9

stack0..stack9

POKER VALUES pokerval, pokervalplayer, pokervalcommon, pcbits,

npcbits

ishandup, ishandupcommon, ishicard, isonepair,

HAND TESTS istwopair, isthreeofakind, isstraight, isflush, isfullhouse, isfourefakind, isstraightflush

isfullhouse, isfourofakind, isstraightflush,

isroyalflush, isfiveofakind

POCKET/COMMON

TESTS

ishipair, islopair, ismidpair, ishistraight, ishiflush

(UN)KNOWN CARDS nouts, ncardsbetter

NHANDS nhands, nhandshi, nhandslo, nhandsti, prwinnow,

prlosnow

nsuited, nsuitedcommon, tsuit, tsuitcommon, nranked, nrankedcommon, trank, trankcommon,

FLUSHES SETS nstraight, nstraightcommon, nstraightfill, STRAIGHTS nstraightfillcommon, nstraightflush,

nstraigntifficommon, nstraigntifusn,

nstraightflushcommon, nstraightflushfill,

nstraightflushfillcommon

rankbits, rankbitscommon, rankbitsplayer,

RANK BITS rankbitspoker, srankbits, srankbitscommon,

srankbitsplayer, srankbitspoker

rankhi, rankhicommon, rankhiplayer, rankhipoker,

RANK HI srankhi, srankhicommon, srankhiplayer,

srankhipoker

ranklo, ranklocommon, rankloplayer, ranklopoker,

RANK LO sranklo, sranklocommon, srankloplayer,

sranklopoker

Table statistics

Table statistics for the last 15 minutes.

Symbol Meaning

floppct percentage of players seeing the flop for the last 15

hands

turnpct percentage of players seeing the turn for the last 15

hands

riverpct percentage of players seeing the river for the last 15

hands

average number of bets preflop for the last 15

hands

tablepfr pfr percentage preflop for the last 15 hands

maxbalance my highest balance during the session

number of hands played this session by this

handsplayed OpenHoldem instance. The value will be 0 after

connection and increase on handreset if the

userchair is known.

handsplayed_headsup number of consecutive hands played headsup,

meant for opponent modelling

ranked list of player balances (includes players not

balance_rankx (x=0-9) currently in hand, and includes currentbet for each

player as well). rank0 has highest balance.

Colourcodes

Colourcodes for player-profiling if a casino is not supported by PokerTracker. Can also be used for easy adaption against regulars and other friends.

Symbol Meaning

colourcode0..colourcode9

The scraped colourcode for playerX, depending on the tablemap region r\$pXcolourcode. All colourcodes will be transformed to the closest web-safe-colour (216 values). This avoids problems with fuzzy casino-displays and 216 values are more easy to be handled by the enduser.

Betting Action

Betting action symbols depend on game-history, They get updated once per orbit at our turn and can therefore only used at our turn; otherwise they are undefined.

Meaning
which chair was the last to raise in round x
which chairs raised in round x
which chairs called in round x
which chairs folded in round x
does the aggressor chair act after me?
preflop position of the userchair (SB=1 BB=2 Early=3 Middle=4 Late=5 Dealer=6)
preflop position of the raiser (SB=1 BB=2 Early=3 Middle=4 Late=5 Dealer=6)
postflop position of the userchair (first=1 early=2 middle=3 late=4 last=5)
returns true if you are first into the pot (first to act or checked to you)
returns bet position of specified chair
returns deal position of specified chair

Logging

Symbol Meaning

When executed, adds your specified text to the

log\$YourTextGoesHere autoplayer-trace and to the white box in the GUI.

The symbol will be evaluated to 1.

For example if you have the following f\$rais-formula,

```
##f$rais##
[ betround == 2 && (nsuited == 3 || nstraight == 3) && log$ItLives ]
```

and you flop a backdoor flush draw, then OpenHoldem will raise and also add the text "ItLives" to the log file, just before the RAIS line. Everytime a log\$ symbol is accessed it will be flagged for logging. You can use this to track which part of your formula is to blame for the action taken by OpenHoldem.

```
87146
       ****** Autoplayer Trace *****
87147
       f$delay = 0.000 [Line 2/119]
87148
       f$rebuy = 0.000 [Line 3/108]
87149
       f$delay = 0.000 [cached]
87150
       f$allin_on_betsize_balance_ratio
87151
       f$betsize_enable_rounding = 1.000
       f$alli = 0.000 [Line 38/47]
87152
87153
        log$test = 1.000
87154
          call = 0.000
oldem (OpenHoldem Poker - No Limit - blinds 5/10)
DI TOM D
 8 9 10 11 12 13 14 15 16 17 18 19
            Hand #: 1
            NL 5/10/20
            Pot: 0
            log$test
```

Note: You want to use the log\$ symbol at the end of a logic block, to make sure it only gets flagged when all the previous statements are true.

As the log\$-symbols modify the autoplayer-trace they require the autoplayer to be engaged.

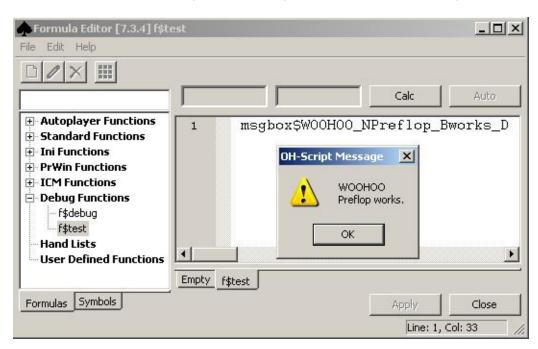
MessageBoxes with Info

In addition to logging it might be useful to notify the user at run-time and pause script-execution, mainly in a debugging session. For this usecase OpenHoldem provides a symbol msgbox\$TEXT, that — surprise — will show a messagebox with text. As the character-set in symbols is slightly limited the following replacements will take place.

- _B will be replaced by a blank
- _C will be replaced by a comma
- _D will be replaced by a dot
- _N will be replaced by a new-line
- _U will be replaced by a underscore

The symbol msgbox\$TEXT will evaluate to 0. An example:

##f\$preflop##
WHEN Others RETURN (Call + msgbox\$W00H00_NPreflop_Bworks_D) FORCE



ICM calculator

An overview of basic ICM push/fold decisions and how they work with OpenHoldem can be found here:

http://www.maxinmontreal.com/wiki/index.php5?title=ICM.

UTG in a symbols name does always mean the first player after BB at a full-ring table with 10 players. If less players do actually sit at the table, the first position (UTG) will be ignored and the first player will be named UTG+1 (or UTG+2, etc.).

The ICM symbols will always assume, that you win against the player with the smallest stack or lose against the player with the biggest stack of the remaining non-folded players (or against N players with the N smallest stacks), so the ICM-symbols will be slightly too pessimistic in multiway-situations. For headsup (including blind-battle) and against concrete players (e.g. icm_alliwinUTG) the symbols are accurate.

Symbol Description

icm my tournament equity before any action is

considered (just balances)

icm_fold my tournament equity if I fold

my tournament equity if I call and win

icm callwin (conservativly: against the smallest remaining

stack)

icm calllose my tournament equity if I call and lose

(conservativly: against the largest remaining stack)

icm_calltie my tournament equity if I call and tie

icm_alliwin0..icm_alliwin9 my tournament equity if I push all-in and win

against 0 - 9 callers

icm allilose0..icm allilose9 my tournament equity if I push all-in and lose

against 0 - 9 callers

icm alliwinSB my tournament equity if I push all-in and win

against the small blind

icm alliwinBB my tournament equity if I push all-in and win

against the big blind

my tournament equity if I push all-in and win

icm_alliwinUTG against UTG

icm_alliwinUTG1	my tournament equity if I push all-in and win against UTG + 1
icm_alliwinUTG2	my tournament equity if I push all-in and win against UTG + 2
icm_alliwinUTG3	my tournament equity if I push all-in and win against UTG + 3
icm_alliwinUTG4	my tournament equity if I push all-in and win against UTG + 4
icm_alliwinUTG5	my tournament equity if I push all-in and win against UTG + 5
icm_alliwinUTG6	my tournament equity if I push all-in and win against UTG + 6 (cut-off)
icm_alliwinCO	my tournament equity if I push all-in and win against the cut-off
icm_alliwinD	my tournament equity if I push all-in and win against the dealer
icm_allitieSB	my tournament equity if I push all-in and tie against the small blind
icm_allitieBB	my tournament equity if I push all-in and tie against the big blind
icm_allitieUTG	my tournament equity if I push all-in and tie against UTG
icm_allitieUTG1	my tournament equity if I push all-in and tie against UTG + 1
icm_allitieUTG2	my tournament equity if I push all-in and tie against UTG + 2
icm_allitieUTG3	my tournament equity if I push all-in and tie against UTG + 3
icm_allitieUTG4	my tournament equity if I push all-in and tie against UTG + 4
icm_allitieUTG5	my tournament equity if I push all-in and tie against UTG + 5
icm_allitieUTG6	my tournament equity if I push all-in and tie against UTG + 6 (cut-off)
icm_allitieCO	my tournament equity if I push all-in and tie

against the cut-off

icm allitieD my tournament equity if I push all-in and win

against the dealer

icm alliloseSB my tournament equity if I push all-in and lose

against the small blind

icm alliloseBB my tournament equity if I push all-in and lose

against the big blind

icm alliloseUTG my tournament equity if I push all-in and lose

against UTG

icm alliloseUTG1 my tournament equity if I push all-in and lose

against UTG + 1

icm alliloseUTG2 my tournament equity if I push all-in and lose

against UTG + 2

icm_alliloseUTG3 my tournament equity if I push all-in and lose

against UTG + 3

icm alliloseUTG4 my tournament equity if I push all-in and lose

against UTG + 4

icm_alliloseUTG5 my tournament equity if I push all-in and lose

against UTG + 5

icm alliloseUTG6 my tournament equity if I push all-in and lose

against UTG + 6 (cut-off)

icm alliloseCO my tournament equity if I push all-in and lose

against the cut-off

icm alliloseD my tournament equity if I push all-in and lose

against the dealer

Hand multiplexor

The purpose of the hand multiplexor is to transfer control to a specific named formula based on your actual 2 card hand. Note that the X's and x's are not case sensitive.

Symbol Meaning

f\$\$X Evaluate the UDF that corresponds to my first card

f\$\$XX Evaluate the UDF that corresponds to my first card

and second card

f\$\$XXx Evaluate the UDF that corresponds to my first card,

second card and suited/unsuited state

The X's in the three symbols will be substituted with your actual hand values at time of evaluation. The first "X" will contain the the rank of your highest hole card, the second "X" will contain the rank of your lowest hole card, and the third "x" will contain either a "s" or "o" depending if your hole cards have the same suit or not. The \$\$ will be replaced with a single dollar sign. For example, if my hole cards are AhKh, then "f\$\$X" will result in "f\$A" upon evaluation. Similarly, "f\$\$XX" would result in "f\$AK" and "f\$\$XXx" would result in "f\$AKs".

OpenHoldem will then evaluate the corresponding user defined function and return that value for the hand multiplexor symbol. If there is not a corresponding UDF defined for a given set of hole cards, then the return result is zero, not an error. The values returned by the hand specific UDFs are entirely defined by you; one simple suggestion is to have them return 1 for call and 2 for raise; if you do this then your raise formula can do this:

|| [f\$\$XXx >= 2] //hand specific formula calc

Memory Symbols

The purpose of these symbols is to provide storage for values in order to reuse them at some later time. They once got invented as a substitute for left-hand-side-expressions (assignments of variables), that did not exist in the WinHoldem-scripting-language, but in reality, this is a bit a kludge???. A much better choice for LHS expressions is Perl or via the user-DLL extension. Please see the {Memory Symbols} section for more information.

see the {Memory Symbols}	section for more information.
Symbol	Meaning
	Stores a value. Example: me_st_abc_123_45 - stores the value "123.45" in variable "abc".Use "_" for the decimal in values. Allowed right hand side expressions are:
me_st_	 numbers, e.g. me_st_Pi_3_141 symbols, e.g. me_st_MyChair_userchair functions, e.g. me_st_CallFunction_f\$call memory symbols, e.g. me st OldValue me re CurrentValue
me_re_	Retrieves a previously stored value. Example: me_re_abc - retrieves the value from variable "abc".
me_inc_	Increments a memory_symbol by 1. It is used mostly for keeping a running counter of events that happen one at a time.

Hand Symbols

You can reference your dealt hand (hole cards) directly by using the \$ symbols. The general form of the \$ hand symbols are: \$RRs. The \$ character is required, followed by 1 or 2 standard card rank characters. The hand symbols are not case sensitive. A,K,Q,J,T,9-2 or by X or x which is a wild card that will match any rank. This is followed by an optional suit indicator: s - suited, o - offsuit. If the suit indicator is omitted then suit does not matter. Examples:

Symbol	Maaning
Symbol	Meaning

\$AA true if you were dealt aces \$JT true if you were dealt jack ten

\$AKs true if you were dealt ace king suited \$490 true if you were dealt four nine offsuit \$QXs true if you were dealt queen any suited

\$J true if you were dealt any jack

\$XXs true if you were dealt any suited hand

\$XXo true if you were dealt any offsuited

hand

\$q2 true if you were dealt queen two

Card Symbols

The actual card value or rank/suit value for both your hand and the board can be referenced using card symbols. The general form of the \$\$ card symbols are: \$\$AB#. The card symbols are not case sensitive. \$\$ is required. A - is p for player, c for common B - is c for card, r for rank, s for suit # - is the card number (player 0-1 for Holdem and 0-3 for Omaha, common 0-4)

Examples

Symbol	Meaning
\$\$pc0	the card value for player card zero (1st dealt card)
\$\$pr3	the rank value of player card three (4th dealt card, Omaha)
\$\$ps0	the suit value for player card zero (1st dealt card)
\$\$cc0	the card value of common card zero (1st common card dealt)
\$\$cr1	the rank value of common card oneo (2nd common card dealt)
\$\$cs4	the suit value of common card four (5th common card dealt - river)

Card Values

The rank values are 2..14 for 2..ace (high), the suit values are 0..3 (indentical to the values of the poker-eval-library) and -1 for undefined. Again the representations of cards are the same as the ones used by the internally used by the poker-eval-library, plus 253..255 for cardback / no-card / undefined.

Ranks

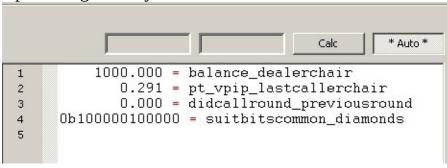
- # CardRank
- 14 Ace
- 13 King
- 12 Queen
- 11 Jack
- 10..2 Ten..Two
- 1 Ace low (used in rankbits of wheels and wheel-draws)
- -1 Undefined (no card or cardback)

Suits

Cardsuit	Value						
Undefined (no card or cardback) -1							
Clubs	2						
Diamonds	1						
Hearts	0						
Spades	3						

Multiplexer

The multiplexer-engine is a module that enables you to access lots of symbols in a poker-logical way.



Historic background

The philosophy of WinHoldem in the by-gone past and early OpenHoldem was: provide technical symbols and let the user do all the rest. For example there were symbols currentbet0..currentbet9 for the bets of the chairs 0,.9. If the user wanted to figure out the currentbet of the big blind he first had to create a function that identified the big-blind-chair

```
##f$bigblindchair##
// Chair 0 dealt 2nd?
ac_dealpos0 == 2 ? 0 :
// Chair 1 dealt 2nd?
ac_dealpos1 == 2 ? 1 :
// Chair 2 dealt 2nd?
ac_dealpos2 == 2 ? 2 :
```

This functions looks for the chair that got dealt 2nd and returns the chair number, assuming it is the big-blind (it ignores some finer details like a missing small-blind or reversed blinds headsup). With this function it is now possible to lookup the currentbet of the big-blind in a similar way:

```
##f$bigblindcurrentbet##
WHEN bigblindchair == 0 RETURN currentbet0 FORCE
WHEN bigblindchair == 1 RETURN currentbet1 FORCE
WHEN bigblindchair == 2 RETURN currentbet2 FORCE
...
WHEN bigblindchair == 9 RETURN currentbet9 FORCE
```

Doable, but somewhat clumsy and lots of work, as such functions had to be created for every logical chair and for every symbol you, the user, wanted to access.

The new multiplexer

First: OpenHoldem provides more than a dozen built-in logical chair-symbols like dealerchair, firstraiserchair, lastcallerchair utgchair. Second: the multiplexerengine makes it possible to use these chair-symbols for indexing numbered symnols. For example you can now write:

```
WHEN currentbet_lastraiserchair > ...
WHEN balance_bigblindchair <= ...
WHEN pt_vpip_smallblindchair < ...</pre>
```

Chair symbols supported

The multiplexer-engine supports the following chair-symbols for indexing other symbols:

- userchair
- bigblindchair
- smallblindchair
- dealerchair
- buttonchair (synonym for dealerchair)
- cutoffchair
- mp3chair
- mp2chair
- mp1chair
- ep3chair
- ep2chair
- ep1chair
- utgchair (under the gun, the player behind big blind, first to act preflop)
- firstraiserchair
- lastraiserchair
- raischair (historic synonym for lastraiserchair)
- firstcallerchair
- lastcallerchair
- headsupchair (chair of your last remaining opponent)
- bigstackchair
- smallstackchair

Betrounds

Besides chair symbols the multiplexer engine supports also betting-round symbols for indexing of game-history symbols. It is now possible to write code like ncurentbets_previousround or the verbose nplayersround_flop instead of nplayersround2. Symbols supported:

- preflop
- flop
- turn
- river
- currentround
- previousround

Suits

Additionally the new symbols suitbitsplayer N = 0..3 and rankbitsplayer can be addressed in a verbose way with the following multiplexer-postfixes.

- clubs
- diamonds
- hearts
- spades

PokerTracker

In the past OpenHoldem supported similar functionality for PokerTracker symbols only. These symbols get now also handled by the multiplexer in order to avoid duplicate functionality in the code-base. as a consequence some symbols got renamed slightly. E.g. pt_vpip_lastcallerchair instead of the former pt_vpip_lastcaller.

OpenPPL and user-defined symbols

Of course the multiplexer-engine works also for OpenPPL. For example the OpenPPL-library contains technical functions StackSize0..Stacksize9 and the values can be addressed in a logical way like StackSize_dealerchair. And of course it works for user-defined functions exactly the same way.

▶Poker Tracker symbols (Version 4)

Assuming all the prerequisites as described in the configuration manual are met, the following native OpenHoldem symbols will be available to your bot for use in its logic processing. The appropriate use of these symbols in opponent modeling is beyond the scope of this document, but plenty of references can be found on the Internet with a Google search.

All the symbols below are available both for cash games and for tournaments (SNG, MTT). OpenHoldem will automatically adapt the database-queries for your game-type and provide the right stats. All stats can be accessed in multiple ways:

- for the chair of the aggressor, e.g. pt_icon_raischair. This situation is one of the most common needs for PokerTracker stats.
- for a single opponent *headsup*, e.g. *pt_icon_headsup*
- to identify certain players by their preflop-position use the postfixes *smallblind*, *bigblind*, *dealer*, *cutoff*, *user*.
- besides of that you can use *firstcaller*, *lastcaller* and *firstraiser*. They work for the current orbit only.

Summary

- You need to use the prefix "pt_".
- Then add the symbol name.
- Finally add a chair number or one of the many postfixes described in the chapter about the multiplexer-engine. The list gets extended over time, here are some examples:
 - _smallblindchair
 - _dealerchair
 - _firstcallerchair
 - _lastraiserchair
 - userchair
 - _headsupchair

General stats

Symbol Meaning

Poker Tracker number of hands that are in the

database

vpip Poker Tracker VP\$IP

pfr Poker Tracker pre-flop raise

wtsd Poker Tracker went to showdown wssd Poker Tracker won \$ at showdown

aggr_factor Poker Tracker total aggression

total_af Poker Tracker total aggression excluding preflop

postflop_fold_to_3bet

Poker Tracker total folded while facing 3bet

excluding preflop

Preflop stats

Symbol Meaning

preflop_aggr_factor Poker Tracker preflop aggression factor

rfi Poker Tracker pre-flop raise first in

3bet Poker Tracker 3bet preflop

fold_to_3bet Poker Tracker folded while facing 3bet preflop

after raising

preflop_called_raise Poker Tracker pre-flop called raise steal_attempt Poker Tracker attempt to steal blinds bb_fold_to_steal Poker Tracker folded big blind to steal sb_fold_to_steal Poker Tracker folded small blind to steal

3bet_vs_steal Poker Tracker 3bet vs. steal bb_3bet_vs_steal Poker Tracker BB 3bet vs. steal sb_3bet_vs_steal Poker Tracker SB 3bet vs. steal

4bet Poker Tracker Preflop 4B

fold_to_4bet Poker Tracker Preflop fold to 4B

fold_to_resteal Poker Tacker Preflop fold when faced 3bet after

attempt to steal blinds

Flop stats

Symbol Meaning

flop_seen Poker Tracker saw flop

flop_af Poker Tracker flop aggression factor

flop_cbet Poker Tracker flop cbet

flop_fold_to_cbet Poker Tracker folded while facing cbet

flop

flop_raise_cbet Poker Tracker raise flop cbet flop_checkraise Poker Tracker flop check-raise flop_donkbet Poker Tracker donk bet flop flop_afq Flop Aggression frequency

flop_float Bet IP flop vs missed cbet

Turn stats

Symbol Meaning

turn_seen Poker Tracker saw turn

turn_af Poker Tracker turn aggression factor

turn_cbet Poker Tracker turn cbet

turn_fold_to_cbet Poker Tracker folded while facing cbet

turn

turn_checkraise Poker Tracker turn Check-Raise turn_checkcall Poker Tracker turn Check-Call

turn_afq Turn Aggression frequency

River stats

Symbol Meaning

river_seen Poker Tracker saw river

river_af Poker Tracker river aggression factor

Poker Tracker folded while facing cbet river_fold_to_cbet

river

Poker Tracker river bet river_bet

River Aggression frequency river_afq

Starting Hand Ranking

OpenHoldem uses a starting hand rank table that is based on the 2652 possible starting two card hands. The original discussion around these hand rankings was developed by Ray Bornert and can be found on the WinHoldem forum(http://forum.winholdem.net/wbb/viewtopic.php?t=94 and http://forum.winholdem.net/wbb/viewtopic.php?t=316). It lists for each possible number of opponents each possible starting hand from best to worst and shows the number of hands equal or better. This table is as follows:

- number of opponents on the X axis (using f\$prwin_number_of_oppoents to be flexible)
- hand rank (1 is best) on the Y axis
- *leading : deal probability less than 1/nplayersdealt *
- trailing*: +50% probability all opponent hands are worse

Hand Rank Table

opponents	9	8	7	6	5	4	3	2	1
1	*AAo*								
1	12	12	12	12	12	12	12	12	12
2	*KKo*								
۷	24	24	24	24	24	24	24	24	24
3	*QQo*								
J	36	36	36	36	36	36	36	36	36
4	*AKs*	*AKs*	*JJo*						
7	44	44	48	48	48	48	48	48	48
5	*JJo*	*JJo*	*AKs*	*AKs*	*AKs*	*TTo*	*TTo*	*TTo*	*TTo*
3	56	56	56	56	56	60	60	60	60
6	*AQs*	*AQs*	*AQs*	*AQs*	*TTo*	*AKs*	*990*	*99o*	*990*
U	64	64	64	64	68	68	72	72	72
7	*KQs*	*KQs*	*KQs*	*TTo*	*AQs*	*AQs*	*AKs*	*AKs*	*880*
,	72	72	72	76	76	76	80	80	84
8	*AJs*	*AJs*	*TTo*	*KQs*	*KQs*	*99o*	*AQs*	*880*	*AKs*
Ø	80	80	84	84	84	88	88	92	92

```
9
      *TTo* *TTo* *AJs* *AJs* *AKo* *KOs* *AKo* *AOs* *77o*
        92
              92
                     92
                            92
                                  108
                                         96
                                                112
                                                             104
                                                      100
      *KJs* *AKo* *AKo* *AKo* *AJs* *AKo* *AJs* *AKo* *AQs*
10
       100
              116
                     116
                           116
                                  116
                                         120
                                                120
                                                      124
                                                             112
                          *KJs*
                                 *990* *AJs* *KOs* *AJs* *AKo*
      *AKo* *KJs*
                   *KJs*
11
       124
              124
                     124
                           124
                                  128
                                         128
                                                128
                                                      132
                                                             136
                          *ATs*
                                 *KJs* *KJs* *880* *770*
      *ATs* *ATs* *ATs*
                                                            *AJs*
12
       132
                                  136
              132
                     132
                           132
                                         136
                                                140
                                                      144
                                                             144
      *QJs*
            *QJs*
                   *OJs*
                          *QJs*
                                 *ATs* *ATs* *ATs* *KOs* *AOo*
13
       140
              140
                     140
                           140
                                  144
                                         144
                                                148
                                                      152
                                                             168
      *KTs* *KTs* *KTs* *990*
                                 *OJs* *AOo* *AOo* *ATs*
                                                            *ATs*
14
                           152
                                  152
                                                             176
       148
              148
                     148
                                         168
                                                172
                                                      160
      *QTs* *QTs* *990* *KTs* *AQo* *880* *KJs* *AQo* *660*
15
                     160
                           160
                                  176
                                         180
                                                180
       156
              156
                                                      184
                                                             188
             *990* *QTs* *AQo* *KTs* *QJs*
                                              *QJs*
      *JTs*
                                                     *KJs* *AJo*
16
       164
                                  184
                                                             212
              168
                     168
                           184
                                         188
                                                188
                                                      192
             *JTs* *AQo* *QTs* *KQo* *KTs* *KTs* *AJo* *KQs*
17
       176
              176
                     192
                           192
                                  208
                                         196
                                                196
                                                      216
                                                             220
      *AQo *AQo* *JTs*
                          *JTs*
                                 *QTs* *KQo* *KTs* *ATo*
18
                     200
                           200
                                         220
                                                      224
       200
              200
                                  216
                                                220
                                                             244
             *KQo *KQo* *KQo* *88o* *QTs* *AJo* *KQo* *A9s*
      *A9s
19
       208
              224
                    224
                           224
                                  228
                                         228
                                                244
                                                      248
                                                             252
                                       *AJo* *77o* *A9s*
                                                            *KJs*
                   *A9s*
      *KQo
             *A9s
                          *880*
                                 *JTs*
20
                                                             260
       232
              232
                     232
                           236
                                  236
                                         252
                                                256
                                                      256
       *880
             *88o
                    *880*
                          *A9s* *AJo*
                                        *JTs*
                                              *OTs* *OJs* *A8s*
21
       244
              244
                     244
                           244
                                  260
                                         260
                                                264
                                                             268
                                                      264
                                 *A9s* *A9s* *ATo* *KTs*
             *K9s
      *K9s
                    *AJo
                          *AJo*
22
              252
                                                             276
       252
                    268
                           268
                                  268
                                         268
                                                272
                                                      288
                          *K9s* *KJo* *KJo* *ATo*
                                                    *660* *KO0*
       *T9s
             *T9s
                    *K9s
23
       260
              260
                    276
                           276
                                  292
                                         292
                                                296
                                                      300
                                                             300
       A8s
             *AJo
                    *A8s
                           *KJo
                                 *K9s* *77o*
                                               *JTs*
                                                    *A8s* *55o*
24
       268
                           300
                                  300
                                               304
              284
                    284
                                         304
                                                      308
                                                             312
                                 *A8s* *ATo* *KJo* *KJo* *A7s*
       J9s
             *A8s
                    *T9s
                           *A8s
25
       276
              292
                     292
                           308
                                  308
                                         328
                                                328
                                                      332
                                                             320
                                 *QJo* *K9s* *A8s* *QTs* *A9o*
       Q9s
              J9s
                    *KJo
                           *Q9s
26
              300
                     316
                           316
                                  332
                                         336
                                                336
                                                             344
       284
                                                      340
```

27	77o	Q9s	*Q9s	*QJo	*ATo	*A8s*	*K9s*	*K9s*	*KJo*
	296	308	324	340	356	344	344	348	368
28	AJo	KJo	J9s	*T9s	*77o	*QJo*	*QJo*	*JTs*	*QJs*
	320	332	332	348	368	368	368	356	376
29	A5s	77o	QJo	*J9s	*Q9s	*Q9s*	*66o*	*A7s*	*K9s*
	328	344	356	356	376	376	380	364	384
30	A7s	A5s	77o	ATo	*T9s	*KTo*	*KTo*	*KTo*	*KTo*
	336	352	368	380	384	400	404	388	408
31	A4s	A7s	A7s	77o	*J9s	*A7s*	*A7s*	*QJo*	*A80*
	344	360	376	392	392	408	412	412	432
32	KJo 368	QJo 384	A5s 384	A7s 400	*KTo 416		*Q9s* 420	*A90* 436	*A6s* 440
33	A3s	A4s	ATo	KTo	*A7s	*T9s	*QTo*	*550*	*QTs*
	376	392	408	424	424	424	444	448	448
34	66o	A3s	A4s	A5s	*A5s	*QTo	*J9s*	*A5s*	*A5s*
	388	400	416	432	432	448	452	456	456
35	A6s	A6s	KTo	A4s	QTo	*A5s	*A5s*	*A6s*	*A4s*
	396	408	440	440	456	456	460	464	464
36	QJo	ATo	A6s	QTo	JTo	*660	*T9s*	*Q9s*	*A70*
	420	432	448	464	480	468	468	472	488
37	K8s	K8s	K8s	K8s	K8s	*JTo	*A90*	*QTo*	*QJo*
	428	440	456	472	488	492	492	496	512
38	A2s	66o	A3s	A6s	A4s	*A6s	*A6s*	*A8o*	*K8s*
	436	452	464	480	496	500	500	520	520
39	T8s	A2s	JTo	JTo	A6s	*K8s	*JTo*	*K8s*	*A3s*
	444	460	488	504	504	508	524	528	528
40	98s	T8s	QTo	A3s	66o	*A4s	*K8s*	*A4s*	*K90*
	452	468	512	512	516	516	532	536	552
41	J8s	KTo	66o	66o	A3s	A9o	*A4s*	*J9s*	*44o*
	460	492	524	524	524	540	540	544	564
42	ATo	98s	T8s	T8s	Q8s	A3s	*55o	*K9o*	*Q9s*
	484	500	532	532	532	548	552	568	572
43	Q8s	Q8s	A2s	Q8s	T8s	Q8s	*A3s	*A3s*	*JTs*
	492	508	540	540	540	556	560	576	580
44	55o	J8s	Q8s	A2s	K7s	K7s	*K7s	*JTo*	*QTo*
	504	516	548	548	548	564	568	600	604

45	K7s	JTo	98s	J8s	J8s	J8s	*Q8s	*K7s*	*A6o*
	512	540	556	556	556	572	576	608	628
46	JTo	QTo	J8s	K7s	98s	T8s	*A8o	*A7o*	*K7s*
	536	564	564	564	564	580	600	632	636
47	KTo	K7s	K7s	98s	A2s	A2s	*K9o	*T9s*	*A5o*
	560	572	572	572	572	588	624	640	660
48	44o	55o	K6s	K6s	A9o	98s	*J8s	*Q8s*	*A2s*
	572	584	580	580	596	596	632	648	668
49	33o	87s	55o	55o	K6s	K9o	*A2s	*A2s*	*K6s*
	584	592	592	592	604	620	640	656	676
50	22o	K6s	87s	A9o	K9o	55o	*T8s	*K6s*	*A4o*
	596	600	600	616	628	632	648	664	700
51	QTo	44o	K5s	87s	55o	K6s	*K6s	*440*	*K8o*
	620	612	608	624	640	640	656	676	724
52	87s	97s	97s	K5s	K5s	A8o	98s	*Q9o*	*Q8s*
	628	620	616	632	648	664	664	700	732
53	K6s	33o	44o	97s	87s	Q9o	Q9o	*A5o*	*J9s*
	636	632	628	640	656	688	688	724	740
54	97s	22o	K4s	Q7s	Q7s	K5s	A7o	*A6o*	*A3o*
	644	644	636	648	664	696	712	748	764
55	K5s	K5s	T7s	K4s	A8o	T9o	K5s	*J8s*	*K5s*
	652	652	644	656	688	720	720	756	772
56	76s	T7s	Q7s	T7s	Q9o	Q7s	J9o	*K5s*	*Q9o*
	660	660	652	664	712	728	744	764	796
57	T7s	K4s	J7s	K9o	T7s	J9o	T9o	*K8o	*JTo*
	668	668	660	688	720	752	768	788	820
58	K4s	76s	K3s	J7s	97s	K4s	Q7s	*T8s	*K7o*
	676	676	668	696	728	760	776	796	844
59	K3s	Q7s	A9o	K3s	K4s	T7s	A5o	*A4o	*A2o*
	684	684	692	704	736	768	800	820	868
60	Q7s	K3s	76s	T9o	T9o	J7s	44o	*J9o	*K4s*
	692	692	700	728	760	776	812	844	876
61	K2s	J7s	33o	44o	J7s	87s	K4s	*Q7s	*33o*
	700	700	712	740	768	784	820	852	888
62	J7s	K2s	K2s	Q6s	J9o	A7o	A6o	*K4s	*Q7s*
	708	708	720	748	792	808	844	860	896

63	86s	86s	Q6s	Q9o	K3s	97s	K8o	*98s	*K6o*
	716	716	728	772	800	816	868	868	920
64	65s	65s	22o	J9o	Q6s	Q6s	J7s	A3o	*T9s*
	724	724	740	796	808	824	876	892	928
65	54s	Q6s	K9o	K2s	44o	K3s	Q6s	K7o	*J8s*
	732	732	764	804	820	832	884	916	936
66	Q6s	54s	86s	76s	K2s	440	T7s	T9o	*K3s*
	740	740	772	812	828	844	892	940	944
67	75s	A9o	T9o	A8o	76s	A5o	97s	Q6s	*Q8o*
	748	764	796	836	836	868	900	948	968
68	Q5s	Q5s	65s	86s	Q5s	K8o	87s	K3s	*Q6s*
	756	772	804	844	844	892	908	956	976
69	96s	T9o	Q5s	Q5s	A7o	K2s	K3s	J7s	*J9o*
	764	796	812	852	868	900	916	964	1000
70	Q4s	96s	J9o	33o	86s	Q5s	A4o	Q8o	*K5o*
	772	804	836	864	876	908	940	988	1024
71	Q3s	75s	Q9o	65s	A5o	A6o	Q5s	A2o	*K2s*
	780	812	860	872	900	932	948	1012	1032
72	64s	Q4s	96s	Q4s	Q4s	76s	K2s	33o	*Q5s*
	788	820	868	880	908	940	956	1024	1040
73	T9o	K9o	54s	96s	96s	A4o	Q8o	K6o	*T8s*
	812	844	876	888	916	964	980	1048	1048
74	Q2s	J9o	Q4s	22o	T6s	86s	K7o	Q5s	*J7s*
	820	868	884	900	924	972	1004	1056	1056
75	A9o	Q3s	A8o	T6s	33o	Q4s	A3o	K2s	*K4o*
	844	876	908	908	936	980	1028	1064	1080
76	T6s	64s	75s	Q3s	K8o	Q8o	J8o	T7s	*Q7o*
	852	884	916	916	960	1004	1052	1072	1104
77	53s	T6s	T6s	A7o	J6s	J6s	Q4s	J8o	*T9o*
	860	892	924	940	968	1012	1060	1096	1128
78	J6s	Q9o	Q3s	J6s	65s	T8o	T8o	97s	*Q4s*
	868	916	932	948	976	1036	1084	1104	1136
79	85s	Q2s	J6s	75s	Q3s	T6s	J6s	87s	*J8o*
	876	924	940	956	984	1044	1092	1112	1160
80	J9o	J6s	Q2s	54s	A6o	96s	76s	Q4s	*K3o*
	900	932	948	964	1008	1052	1100	1120	1184

81	K9o	85s	64s	Q2s	A4o	J8o	A2o	K5o	*22o*
	924	940	956	972	1032	1076	1124	1144	1196
82	43s	53s	85s	A5o	T8o	A3o	98o	T8o	*Q6o*
	932	948	964	996	1056	1100	1148	1168	1220
83	J5s	A8o	J5s	J5s	J5s	K7o	33o	J6s	*Q3s*
	940	972	972	1004	1064	1124	1160	1176	1228
84	Q9o	J5s	53s	K8o	Q8o	Q3s	K6o	Q3s	*98s*
	964	980	980	1028	1088	1132	1184	1184	1236
85	74s	J4s	A7o	85s	Q2s	98o	T6s	Q7o	*T7s*
	972	988	1004	1036	1096	1156	1192	1208	1244
86	J4s	74s	J4s	64s	75s	33o	86s	98o	*J6s*
	980	996	1012	1044	1104	1168	1200	1232	1252
87	J3s	43s	A5o	A4o	J8o	J5s	96s	K4o	*K2o*
	988	1004	1036	1068	1128	1176	1208	1256	1276
88	J2s	J3s	J3s	T8o	98o	65s	Q3s	T6s	*Q2s*
	996	1012	1044	1092	1152	1184	1216	1264	1284
89	95s	95s	95s	J4s	54s	Q2s	J5s	J5s	*Q5o*
	1004	1020	1052	1100	1160	1192	1224	1272	1308
90	63s	J2s	74s	A6o	22o	A2o	Q2s	Q2s	*J5s*
	1012	1028	1060	1124	1172	1216	1232	1280	1316
91	A8o	63s	T8o	98o	A3o	75s	K5o	96s	T8o
	1036	1036	1084	1148	1196	1224	1256	1288	1340
92	T5s	T5s	J2s	Q8o	K7o	J4s	J4s	Q6o	J7o
	1044	1044	1092	1172	1220	1232	1264	1312	1364
93	52s	A5o	43s	J8o	J4s	K6o	65s	76s	Q4o
	1052	1068	1100	1196	1228	1256	1272	1320	1388
94	42s 1060	A7o 1092	A4o 1124	J3s 1204	85s 1236	85s 1264	-		97s 1396
95	T4s	T8o	K8o	53s	J3s	54s	75s	J7o	J4s
	1068	1116	1148	1212	1244	1272	1304	1352	1404
96	T3s	T4s	T5s	A3o	64s	J3s	J7o	22o	T6s
	1076	1124	1156	1236	1252	1280	1328	1364	1412
97	84s	98o	98o	95s	95s	22o	J3s	K3o	J3s
	1084	1148	1180	1244	1260	1292	1336	1388	1420
98	98o	T3s	J8o	J2s	A2o	T5s	85s	J4s	Q3o
	1108	1156	1204	1252	1284	1300	1344	1396	1444

99	T2s	84s	T4s	T5s	T5s	95s	K4o	T7o	98o
	1116	1164	1212	1260	1292	1308	1368	1420	1468
100	A5o	52s	63s	74s	J2s	J2s	T7o	Q5o	T7o
	1140	1172	1220	1268	1300	1316	1392	1444	1492
101	T8o	A4o	A6o	K7o	K6o	64s	T5s	J3s	J6o
	1164	1196	1244	1292	1324	1324	1400	1452	1516
102	A7o	T2s	A3o	T4s	T4s	K5o	95s	K2o	87s
	1188	1204	1268	1300	1332	1348	1408	1476	1524
103	73s	42s	Q8o	43s	53s	Q7o	97o	T5s	J2s
	1196	1212	1292	1308	1340	1372	1432	1484	1532
104	32s	K8o	T3s	A2o	74s	87o	87o	97o	96s
	1204	1236	1300	1332	1348	1396	1456	1508	1540
105	A4o	A3o	84s	T3s	87o	T4s	22o	87o	Q2o
	1228	1260	1308	1340	1372	1404	1468	1532	1564
106	94s	J8o	T2s	84s	T3s	T7o	Q60	65s	J5o
	1236	1284	1316	1348	1380	1428	1492	1540	1588
107	93s	A6o	52s	63s	97o	J7o	J2s	J2s	T5s
	1244	1308	1324	1356	1404	1452	1500	1548	1596
108	62s	73s	A2o	T2s	43s	97o	54s	95s	T4s
	1252	1316	1348	1364	1412	1476	1508	1556	1604
109	A3o	Q8o	K7o	87o	T7o	74s	T4s	75s	97o
	1276	1340	1372	1388	1436	1484	1516	1564	1628
110	K8o	94s	42s	K6o	T2s	K4o	K3o	Q4o	J4o
	1300	1348	1380	1412	1444	1508	1540	1588	1652
111	J8o	32s	87o	52s	84s	T3s	64s	T4s	T6o
	1324	1356	1404	1420	1452	1516	1548	1596	1676
112	92s 1332	93s 1364	94s 1412	94s 1428	K5o 1476	53s 1524	Q5o 1572		86s 1684
113	A6o	A2o	73s	97o	Q7o	Q6o	T3s	J6o	95s
	1356	1388	1420	1452	1500	1548	1580	1628	1692
114	87o	87o	93s	93s	63s	84s	K2o	T3s	T3s
	1380	1412	1428	1460	1508	1556	1604	1636	1700
115	Q8o	92s	92s	73s	J7o	T2s	74s	Q3o	J3o
	1404	1420	1436	1468	1532	1564	1612	1660	1724
116	83s	62s	32s	42s	94s	K3o	84s	54s	76s
	1412	1428	1444	1476	1540	1588	1620	1668	1732

117	A2o	K7o	97o	T7o	K4o	94s	T2s	T6o	87o
	1436	1452	1468	1500	1564	1596	1628	1692	1756
118	82s	83s	K6o	K5o	93s	43s	76o	J5o	T2s
	1444	1460	1492	1524	1572	1604	1652	1716	1764
119	97o	97o	62s	92s	52s	63s	Q4o	T2s	96o
	1468	1484	1500	1532	1580	1612	1676	1724	1788
120	72s	82s	83s	Q7o	73s	76o	94s	96o	J2o
	1476	1492	1508	1556	1588	1636	1684	1748	1812
121	K7o	76o	T7o	J7o	Q6o	Q5o	53s	64s	85s
	1500	1516	1532	1580	1612	1660	1692	1756	1820
122	76o	72s	82s	32s	76o	93s	J6o	86o	T5o
	1524	1524	1540	1588	1636	1668	1716	1780	1844
123	T7o	K6o	760	76o	92s	K2o	T6o	76o	94s
	1548	1548	1564	1612	1644	1692	1740	1804	1852
124	65o	T7o	K5o	83s	K3o	860	860	Q2o	T4o
	1572	1572	1588	1620	1668	1716	1764	1828	1876
125	K6o	65o	J7o	62s	42s	92s	96o	94s	75s
	1596	1596	1612	1628	1676	1724	1788	1836	1884
126	86o	K5o	72s	K4o	860	73s	93s	74s	93s
	1620	1620	1620	1652	1700	1732	1796	1844	1892
127	54o	86o	Q7o	82s	83s	96o	Q3o	84s	86o
	1644	1644	1644	1660	1708	1756	1820	1852	1916
128	K5o	J7o	K4o	860	K2o	52s	63s	J4o	65s
	1668	1668	1668	1684	1732	1764	1828	1876	1924
129	J7o	54o	860	Q6o	Q5o	Q4o	43s	93s	95o
	1692	1692	1692	1708	1756	1788	1836	1884	1948
130	Q7o	Q7o	65o	K3o	82s	J6o	J5o	53s	T3o
	1716	1716	1716	1732	1764	1812	1860	1892	1972
131	75o	K4o	K3o	72s	62s	T6o	92s	J3o	84s
	1740	1740	1740	1740	1772	1836	1868	1916	1980
132	K4o	K3o	Q6o	K2o	32s	83s	73s	92s	92s
	1764	1764	1764	1764	1780	1844	1876	1924	1988
133	K3o	750	K2o	65o	96o	42s	Q2o	T5o	76o
	1788	1788	1788	1788	1804	1852	1900	1948	2012
134	96o	K2o	540	96o	65o	65o	65o	63s	T2o
	1812	1812	1812	1812	1828	1876	1924	1956	2036

135	64o	Q6o	96o	Q5o	T6o	82s	83s	65o	74s
	1836	1836	1836	1836	1852	1884	1932	1980	2044
136	K2o	96o	750	54o	Q4o	Q3o	J4o	73s	85o
	1860	1860	1860	1860	1876	1908	1956	1988	2068
137	53o	640	Q5o	T6o	J6o	J5o	52s	43s	54s
	1884	1884	1884	1884	1900	1932	1964	1996	2076
138	Q6o	Q5o	T6o	750	72s	62s	750	95o	64s
	1908	1908	1908	1908	1908	1940	1988	2020	2084
139	85o	T6o	Q4o	Q4o	750	750	82s	750	83s
	1932	1932	1932	1932	1932	1964	1996	2044	2092
140	T6o	53o	64o	J6o	Q3o	32s	850	83s	94o
	1956	1956	1956	1956	1956	1972	2020	2052	2116
141	Q50	Q4o	J6o	Q3o	540	Q2o	42s	J2o	75o
	1980	1980	1980	1980	1980	1996	2028	2076	2140
142	43o	850	Q3o	Q2o	J5o	72s	J3o	850	82s
	2004	2004	2004	2004	2004	2004	2052	2100	2148
143	Q4o	J6o	850	850	Q2o	J4o	T5o	T4o	93o
	2028	2028	2028	2028	2028	2028	2076	2124	2172
144	Q3o	Q3o	53o	64o	85o	54o	62s	82s	73s
	2052	2052	2052	2052	2052	2052	2084	2132	2180
145	740	Q2o	Q2o	J5o	J4o	850	950	52s	65o
	2076	2076	2076	2076	2076	2076	2108	2140	2204
146	Q2o	43o	J5o	53o	64o	95o	54o	T3o	53s
	2100	2100	2100	2100	2100	2100	2132	2164	2212
147	J6o	740	740	J40	95o	T5o	72s	540	63s
	2124	2124	2124	2124	2124	2124	2140	2188	2220
148	63o	J5o	43o	950	J3o	J3o	T4o	62s	84o
	2148	2148	2148	2148	2148	2148	2164	2196	2244
149	J5o	J4o	J4o	J3o	T5o	64o	J2o	42s	92o
	2172	2172	2172	2172	2172	2172	2188	2204	2268
150	95o	95o	950	740	53o	J2o	32s	64o	43s
	2196	2196	2196	2196	2196	2196	2196	2228	2276
151	52o	63o	J3o	T5o	J2o	T4o	64o	72s	74o
	2220	2220	2220	2220	2220	2220	2220	2236	2300
152	J4o	J3o	J2o	43o	740	740	T3o	T2o	72s
	2244	2244	2244	2244	2244	2244	2244	2260	2308

153	42o	J2o	63o	J2o	T4o	53o	74o	74o	54o
	2268	2268	2268	2268	2268	2268	2268	2284	2332
154	J3o	52o	T5o	T4o	430	T3o	840	94o	64o
	2292	2292	2292	2292	2292	2292	2292	2308	2356
155	J2o	T5o	52o	63o	T3o	840	T2o	840	52s
	2316	2316	2316	2316	2316	2316	2316	2332	2364
156	840	840	T4o	T3o	840	T2o	53o	32s	62s
	2340	2340	2340	2340	2340	2340	2340	2340	2372
157	T5o	420	840	840	63o	940	940	93o	83o
	2364	2364	2364	2364	2364	2364	2364	2364	2396
158	32o	T4o	T3o	52o	T2o	43o	93o	53o	82o
	2388	2388	2388	2388	2388	2388	2388	2388	2420
159	T4o	T3o	420	T2o	940	63o	63o	92o	42s
	2412	2412	2412	2412	2412	2412	2412	2412	2428
160	T3o	32o	T2o	94o	52o	93o	430	63o	73o
	2436	2436	2436	2436	2436	2436	2436	2436	2452
161	73o	T2o	73o	42o	93o	73o	92o	43o	53o
	2460	2460	2460	2460	2460	2460	2460	2460	2476
162	T2o	73o	94o	73o	73o	92o	73o	73o	63o
	2484	2484	2484	2484	2484	2484	2484	2484	2500
163	62o	94o	32o	93o	42o	52o	83o	83o	32s
	2508	2508	2508	2508	2508	2508	2508	2508	2508
164	94o	62o	93o	92o	92o	83o	52o	82o	43o
	2532	2532	2532	2532	2532	2532	2532	2532	2532
165	93o	93o	62o	32o	83o	42o	82o	52o	72o
	2556	2556	2556	2556	2556	2556	2556	2556	2556
166	92o	92o	92o	83o	62o	82o	42o	62o	52o
	2580	2580	2580	2580	2580	2580	2580	2580	2580
167	83o	83o	83o	62o	32o	62o	62o	42o	62o
	2604	2604	2604	2604	2604	2604	2604	2604	2604
168	82o	82o	82o	82o	82o	32o	72o	72o	42o
	2628	2628	2628	2628	2628	2628	2628	2628	2628
169	72o	72o	72o	72o	72o	72o	32o	32o	32o
	2652	2652	2652	2652	2652	2652	2652	2652	2652

Using handrank

The rank order of hands changes depending on the number of opponents (f\$prwin_number_of_oppoents).

From this hand rank table, OpenHoldem will derive the symbols handrank169, handrank1326, handrank2652, handrank1000, and handrankp.

There are two initial methods you can use to leverage these hand rankings in your formula. First, if you want a static rule that will not be affected by folding opponents then you would use this code snippet:

```
[ betround==1 && handrank2652 <= 2652*dealposition/nplayersdealt**2</pre>
```

Second, if you want a dynamic rule that will be affected by folding opponents then you would use this rule:

```
[ betround==1 && handrank2652 <= 2652*betposition/nplayersplaying**2</pre>
```

The table above also flags "good" hands with leading and trailing asterisks. The function used to determine the left asterisk is:

```
handrank2652 <= 2652/nplayersdealt
```

And the function used for the right asterisk is:

```
handrank2652 <= 2652 * (1.0 - e^{**}(ln(0.500)/nopponents))
```

The right asterisk is the point where the probability that all opponent hands are worse becomes 50% or more.

OH-Script

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:35:48.890000

JOH-Script

Introduction

OpenHoldem's \(\pm\)scripting language is very similar to the C programming expression syntax. If you already know and understand C programming then you have an advantage in that the OH scripting language should be very familiar. This section is largely based off of Ray Bornert's original WinHoldem "\(\pm\)C-tutorial" (\(\text{http://www.winholdem.net/help/help-c.html\)).

<u>↓</u>Syntax

The general syntax for OpenHoldem formulas is same as that of the C programming language. However, the formulas are strict *r-values* (right side of an assignment expression), meaning that there is no assignment operator.

Comments

C-syntax means of cource C(++)-comments, too. Comments will improve your code and will make it more easy to read and understand, both for yourself and for other people. Make use of them!

```
// This is a C-style-comment
// till the end of the line.
/*
   and this is a C++-style
   multi-line-comment.
*/
```

Data Types

OH-Script is a programming language without typing. All values are stored as (double) floating point values on the expression stack, regardless if they are \downarrow floating points, \downarrow integers, \downarrow booleans (\downarrow logicals) or \downarrow bit-vectors. For bitwise operations the values are first internally converted to integer values and then the operator is applied.

If a value is interpreted as numerical or boolean depends solely on the context. A common beginners mistake is to mix up boolean and numerical values, boolean and numerical operators.

↓Rounding Errors

One of the main problems with floating-points are the inevitable rounding errors (http://floating-point-gui.de/).. Rounding errors occur for numerical calculations, but also for conversions of floating-point-numbers from ↓decimal format (OH-script) to the internal ↓binary representation (http://en.wikipedia.org/wiki/Binary_numeral_system).

Rounding errors are especially annoying when you compare two numbers, that should be equal from an exact mathematical point of view but are only nearly in practice.

To overcome this problem OpenHoldem uses approximative comparision for relational and equality operators (< < > >= = !=). It will treat two numbers as "equal", when they differ by not more than 10^{-6} . This applies of course only to OH-script; \downarrow DLL--users have to care about correct rounding themselves!

<u>J</u>Boolean Logic

A boolean expression is composed of logical operators (not, and, or, xor) and operands. All boolean operands have only two values - true and false. Each logical operand has a very well defined operation upon the operand(s) in the expression, with a very well defined result. When any numeric value is used in conjunction with a logical operator, any zero values are considered to be false (0) and any non-zero values are considered to be true (1). If the numeric value in question is not zero then it is considered to be true for boolean purposes. Reams of material have been written on boolean logic (http://en.wikipedia.org/wiki/Boolean logic).

Uperators

These operators are listed in order of precedence.

Category	Operator(s)	Associativity
Exponentiation	** ln (not standard ANSI C)	Right to Left
Unary	! ~ - '	Right to Left
Multiplicative	* / %	Left to Right
Additive	+ -	Left to Right
Bitwise Shift	<< >>	Left to Right
Relational	<><=>=	Left to Right
Equality	== != ~~	Left to Right
Bitwise	AND &	Left to Right
Bitwise	XOR ^	Left to Right
Bitwise	OR	Left to Right
Logical	AND &&	Left to Right
Logical	XOR ^^ (not standard ANSI C)	Left to Right
Logical	OR	Left to Right
Conditional	?:	Right to Left
Group	() [] {} (not standard ANSI C)	Left to Right

- Exponentiation Power ** (not ANSI-C) a ** b Standard algebraic exponentiation on a and b. a is raised to the power of b.
- Natural Log ln (not ANSI-C) ln a Standard algebraic natural log of a a == e
 ** (ln a)
- Natural Log Base e (not ANSI-C) e == 2.71828182845905
- Unary A unary operator takes a single operand.
- Logical NOT! False when the operand is true. True when the operand is false. a!a false true true false 01
- Bitwise NOT ~ Logical NOT operation on a bit by bit basis.

• Negation - A good example is the minus sign when it is used to alter the

- sign of a value. Example: "a + b / -4" The minus sign in front of the 4 is a unary minus.
- Bit Count ' (not ANSI-C) Provides a count of the number of bits set in any integer value. Here are 3 examples of the number of bits that are set in some 32-bit integer numbers.

- Multiplicative Multiply * a * b Standard algebraic multiplication on a and b.
- Divide / a / b Standard algebraic division on a and b.
- Modulo % a % b Standard algebraic modulo on a and b.
- Percentage % a % b The result will be (a / 100) * b. The percentage operator got introduced with OpenPPL. OpenHoldem will assume that "%" inside an OH-script function means the modulo-operator and that "%" inside an OpenPPL-function (with WHEN-conditions) means the percentage-operator.
- Additive Add + a + b Standard algebraic addition on a and b.
- Subtract a b Standard algebraic subtraction on a and b.
- Bitwise Shift Left << Slides the entire bit pattern to the left by N bits. Note that the leftmost bits are simply dropped and that the rightmost bits are filled with 0's. The shift magnitude is used as modulo 32, meaning that any shift N that is specified in excess of 32 bits has a N%32 operation performed prior to the shift.

expression binary result a 1100100101110101010101011111010 a<<1</td> 1001001011101010101010111110100 a<<7</td> 101110101011001010111110100000000 a<<31</td> 00000000000000000000000000000000000 a<<32</td> 110010010111010101010101011111010

Bitwise Shift Right >> Slide the entire bit pattern to the right by N bits.
Note that the rightmost bits are dropped and that the leftmost bits are filled with 0's. The shift magnitude is used as modulo 32, meaning that any shift N that is specified in excess of 32 bits has a N%32 operation performed prior to the shift. expression binary result.

expression	binary result
a	110010010111010101100101011111010
a>>1	01100100101110101011001010111101
a>>7	00000001100100101110101011001010
a>>31	000000000000000000000000000000000000000
a>>32	11001001011101010110010101111010

- Relational Less Than < a b a < b -1 0 true 0 0 false +1 0 false
- Greater Than > a b a > b -1 0 false 0 0 false +1 0 true
- Less Than Or Equal <= a b a <= b -1 0 true 0 0 true +1 0 false
- Greater Than or Equal \geq a b a \geq b -1 0 false 0 0 true +1 0 true
- Equality Equal == a b a == b -1 0 false 0 0 true +1 0 false
- Not Equal != a b a != b -1 0 true 0 0 false +1 0 true
- Approximately Equal ~~ this operator rounds both sides to integer numbers before comaring. It is especially to support betsize and stacksize comparisons of converted PPL profiles, as Shanky PPL works with integers only.
- Bitwise AND & Logical AND operation on a bit by bit basis.

expression

binary

- a 11001001011101010110010101111010
- b 01001010010100101110001010101111
- a&b 01001000010100000110000000101010
- Bitwise OR | Logical OR operation on a bit by bit basis.

expression

binary

- a 11001001011101010110010101111010
- b 01001010010100101110001010101111
- a | b 11001011011101111110011111111111
- Bitwise XOR ^ Logical XOR operation on a bit by bit basis.

expression

binary

- a 11001001011101010110010101111010
- b 01001010010100101110001010101111
- $a^{\wedge}b \qquad 10000011001001111000011111010101$
- Logical AND && False when any operand is false. True when both operands are true.
 - a b a && b false false

false true false true false false true true true

• Logical XOR $^{\wedge \wedge}$ (not ANSI-C) False when operands are boolean equal. True when operands are not boolean equal. Note that "a $^{\wedge \wedge}$ b" is equivalent to "(a!=0) $^{\wedge}$ (b!=0)".

a b a ^^ b false false false true true true true false true

• Logical OR || False when both operands are false. True when any operand is true.

a b a || b false false false true true true true true true

• Conditional ?: Standard algorithmic if then else. "a ? b : c" means "If a then b else c".

a b c a?b:c true any any b false any any c

• Grouping Operators ()[]{} Note that [] and {} are not ANSI-C. These grouping operators are used to either visibly separate sections of your code for readability and maintainability purposes, or to affect the precedence of logical operation. Note that unlike WinHoldem's grouping syntax, there is no limitation on how the various grouping operators can be nested.

Numeric Constants Floating point constants

All numeric constants are treated internally as double floating point values in base 2. Floating Point Numeric Constants 123.456 0.987 192837465 .5 17. 5.4321e-76

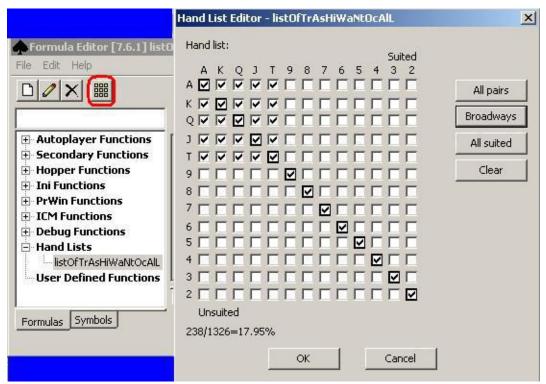
Integer constants There are 4 integer options available as well that allow you to select the base of the constant. The 4 available bases are: 16, 8, 4, 2. Prefixing a numeric constant with a zero followed by a letter (see table below) will specify the numeric base of the constant.

Letter Base Name x 16 Hex o 8 Octal (non-standard ANSI C) q 4 Quadal (non-standard ANSI C) b 2 Binary (non-standard ANSI C) Examples:

Decimal	Hex	Octal	Quadal	Binary
0	0x0	000	0q0	0b0
1	0x1	0o1	0q1	0b1
2	0x2	0o2	0q2	0b10
3	0x3	0o3	0q3	0b11
4	0x4	004	0q10	0b100
5	0x5	0o5	0q11	0b101
6	0x6	006	0q12	0b110
7	0x7	0o7	0q13	0b111
8	0x8	0o10	0q20	0b1000
9	0x9	0o11	0q21	0b1001
10	0xA	0o12	0q22	0b1010
11	0xB	0o13	0q23	0b1011
12	0xC	0o14	0q30	0b1100
13	0xD	0o15	0q31	0b1101
14	0xE	0o16	0q32	0b1110
15	0xF	0o17	0q33	0b1111
16	0x10	0o20	0q100	0b10000
31	0x1F	0o37	0q133	0b11111
127	0x7F	0o177	0q1333	0b1111111
255	0xFF	0o377	0q3333	0b11111111

↓Hand Lists

OpenHoldem Hand List Editor



The OpenHoldem Hand List editor is displayed when you click on the "3x3 yellow squares" toolbar button from the Formula Editor window. This provides a graphical interface for defining the Hand Lists as used within OpenHoldem for various symbols and functions. The output of this window is not all that difficult to understand, and if you choose your card pairs here, save your formula, then open the formula in a text editor, you can quite easily see the format used. On this window, you specify which of your 169 starting hands (hole cards) are included in which Hand List. In the screen shot above, Hand List 7 was selected prior to clicking the "3x3 yellow squares" toolbar button. There is a diagonal group of starting hands running from top left to bottom right and identified with bolded checkboxes. These are the paired starting hands, i.e. AA, KK, JJ, 55. Above this diagonal line are checkboxes for the suited starting hands, i.e. AhKh, 3s2s, Jc7c. Below this diagonal line are checkboxes for the unsuited starting hands, i.e. AhJc, 3d2h.

When the "OK" button is clicked, the starting hand selections you have made are

saved to the Hand List you selected.

Using Hand Lists

Hand lists are used by the following calculated symbols, and can be referenced in OH-Script code, or DLL-based logic.

Symbol Meaning

listXYZ true if your hand is in the list named listXYZ

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-03T20:03:29.062000

OH-Script Explained

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-23T11:38:52.125000

PrWin and Versus - Winning Probabilities

	prwin	versus	weighted prwin
preflop/postflop	both	both	both
accuracy	<pre>depends on f\$prwin_number_of_iterations</pre>	100% exact	depends on f\$prwin_number_
speed	depends on f\$prwin_number_of_iterations	superfast (precomputed table)	depends on f\$prwin_number_
handranges	random only (100% range)	arbitrary handlists, but usually fixed	userdefined range simplified model
opponnents	1-22, using f\$prwin_number_of_opponents	1	1-13, using f\$prwin_number_ (weighted prwin v disabled, if value
formula / DLL	all	all	all

Standard PrWin

Standard PrWin was the first way to calculate winning probabilities with WinHoldem. Just like so many other features of WinHoldem it is based on an oversimplified model, assuming a 100% range for all opponents, i.e. just random cards, no matter what.

This method has an obvious drawback: the calculated winning-percentages are way too optimistic.

Instead of improving that basic model a rude workaround got introduce: f\$prwin_number_of_opponents, a function to adapt the number of opponents considered by the PrWin-Iterator. So instead of modelling hand-ranges for your opponents depending on the texture of the boards and the bettinglevel you had to do some guesswork about how many opponents to add to get the same numbers. For an explanation about the required number of iterations see the chapter "Iterations and Standard Deviation". But as a rule of thumb 10.000 to 100.000 should give you reasonable numbers.

Versus

Versus is a capability introduced into WinHoldem in 2005 (and implemented in OH from startup) which allows the user to assess his probability of winning against a single opponent with a specified handlist. It does use large precomputed tables and the result is substantially obtained by table lookup rather than the Monte Carlo iteration used for Prwin. It is much faster than Prwin, and the user can weight the result by selecting a suitable handlist for the opponent. It is thus a powerful tool for calculating good win probability against a single opponent.

Especially useful is the multiplexer-function

```
vs$multiplex$X$prwin // or prlos, prtie
```

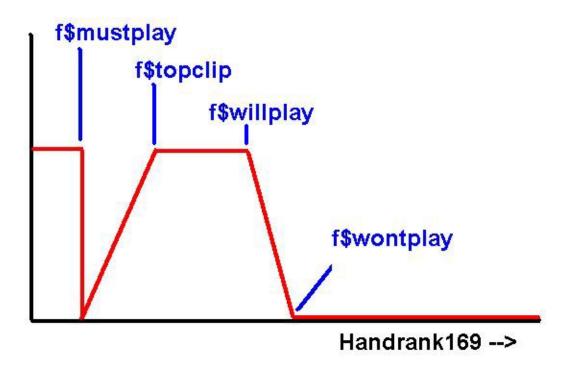
that looks up a numbered list depending on a range-function. For example the calculation of allin-equity could become as simple as

```
##f$allin_equity_against_bigblind##
vs$multiplex$f$bigblind_calling_range$prwin
+ 0.5 * vs$multiplex$f$bigblind_calling_range$prtie
```

Against multiple opponents its utility is much reduced, since the existence of the same cards in different opponent handlists is not allowed for the versus calculation, and combining the results from several independent versus assessments creates an increasingly inaccurate estimate as the number of opponents increases.

Weighted PrWin

Weighted Prwin does work similar to Standard PrWin, but enables you to model opponents handranges. It got developed by Matrix as part of his DLL and then contributed to OpenHoldem.



The hand-range of your opponent gets described by four parameters, each specifying a handrank169-value:

- f\$mustplay: absolute premium-hands, that will never be folded by your opponent, even if the betting-action does not indicate a monster-hand. If you are dead-sure, that your opponent doesn't slowplay aces, you can set f\$mustplay to zero.
- f\$topclip: the top of your opponents range, which will be played with certainty. Cards between f\$topclip and f\$mustplay will also be included, but with reduced probability.
- f\$willplay: the bottom of your opponents range, which will be played with certainty.
- f\$wontplay: the border to pure trash. Hands rated lower will never be played by your opponent. Hands between f\$willplay and f\$wontplay will be included in his range, but with reduced probability.

Ideally you would base the calculations of these points on the statistics for opponent betting pre-flop at various bet levels. This information is not currently available from OH (unless you use the PokerTracker capabilities), but the single most useful statistic, SingleMalt's floppct, can be used. Using this to set f\$willplay and f\$wontplay gives the major advantage which a weighted prwin gives; the elimination of rags from the iterator calculations. A starting point

might be something like:

```
##f$willplay##
handsplayed<10 ? 72 : 180*floppct
##f$wontplay##
f$willplay * 1.4</pre>
```

Weighted PrWin will fall back to Standard PrWin, when the parameters provided are unusable, e.g.

- if f\$willplay is absent, or returns zero.
- if f\$prwin_number_of_opponents is greater than 13.

The calculation will furthermore not weight the cards for an opponent who is BB and has limped into the flop. f\$willplay will be increased internally if it is so low that the iterator cannot assign cards for all opponents from the available matching hands.

Weighted PrWin gives far better results than Standard PrWin, but the ability to specify handranges is still somewhat limited due to the ordering of hands according to their handrank-value. It is e.g. a bit hard to model playing tendencies or the preference for hands with low equity but good playability, like low pairs and suited connectors.

Furthermore the same range will be assumed for all opponents in the hand. As the hand-ranges are more concrete than for Standard PrWin f\$prwin_number_of_opponents should obviously return the correct number of opponents in the hand.

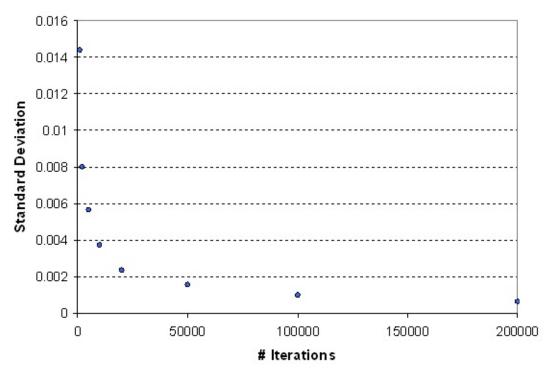
Enhanced PrWin

To overcome that limitations Enhanced PrWin got developed (again by Matrix). Enhanced PrWin does allow you to specify a probability for every single hand and thus gives you exact winning probabilities (provided you model the hand-ranges correctly). However — this mightiness causes some complexity and some labour. Advanced opponent modelling and calculating probabilities for every single hand is too complex for OH-script. Therefore this features is only available at the DLL-level and gets described in more detail in the chapter about DLL-programming.

27T00:12:55.125000

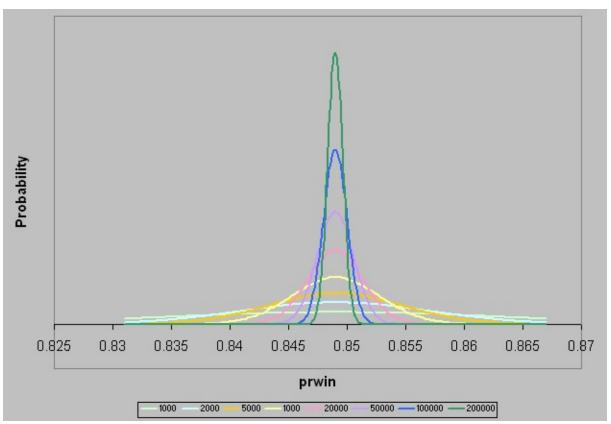
Iterations and Standard Deviation

Running AA headsup preflop with X iterations in OpenHoldem to determine the standard deviation in prwin for the various numbers of iterations. Here are the results:



Here is how to use the data in the above graphs:

- 1. Decide what error you find acceptable 99% of the time. For example, if you are happy with getting prwin's of 0.846 0.852 99% of the time for AA headsup preflop, your error would be (0.849) +- 0.003.
- 2. That error is "3 standard deviations". To get 1 standard deviation, divide by 3 (to give std. dev. = 0.001).
- 3. Find that standard deviation in the top chart and read off the number of iterations required. In our case, it would be 100,000 required iterations.



The prwin values you obtain will be within 1 standard deviation 67% of the time, within 2 standard deviations 95% of the time, and within 3 standard deviations 99% of the time.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-30T22:26:54.921000

ICM - The independent chip model

(This explanation has been provided by an unknown author for the Wiki: http://www.maxinmontreal.com/wiki/index.php5?title=ICM))
Here follows an overview of basic ICM push fold decisions and how they work with OpenHoldem.

Equity

This is obviously very important for botters - and all poker players in fact - to know about. When applied to a pot, it is how much of the pot you currently "own". Say I'm in a \$30 pot heads-up, and neither player has looked at their cards. I could say I'm entitled to half the pot. Now let's take into account your probability of winning. So say we have AKs and we're up against JTo. Our prwin is 0.65, so I should rightly feel I'm entitled to more than half the pot. We could say we currently own 0.65 of the pot (0.65*30). This is our current "pot equity".

Equity doesn't only apply to pots, it applies to tournaments (and to folding). If pot equity is the portion of the pot we currently own, then tournament equity is the portion of the tournament prize pool we currently "own". So if it's a tournament of 2 people, with a prize pool of \$1 and we are equal in chips then I currently "own" half of the prize pool or \$0.5. My tournament equity is 0.5. This is what the OH symbol "icm" will calculate for you: "my tournament equity before any action is considered (just balances)". We thank OH for this, because once you have more than 2 players and the prize structure is complicated then calculating the value for "icm" gets harder.

And we need to know what will happen to our tournament equity when my balance changes from \$0.5 to something else. If I fold, I can lose my blind so it will go down (or stay the same if I didn't put in a blind). If I go all-in it could double, go down to 0, or increase slightly as I could win the blinds. And if I call an all-in it could double or go to zero. Working out what will happen to the balances in such situations - and the resulting value for my tournament equity - is all handled by OH. Again, praise be to OH, because it gets complicated to calculate with more players.

So, using OH we can work out (using the ICM-calculator-symbols):

• icm_fold - my tournament equity (TE) if I fold

- icm_callwin my TE if I call and win
- icm_calllose my TE if I call and lose
- icm_alliwin0 my TE if I go all-in and everyone else folds
- icm_allilose1 my TE if I go all in and lose (against 1 person)
- icm_alliwin1 my TE if I go all in and win (against 1 person)

That's against 1 person. So like our example of fighting over the \$1 prize pool. Or it could be when you're in the BB and calling a SB push or in the SB and pushing against the BB. I haven't got to the multiple people stage yet. More on that later.

You need to set the distribution of the prize pool in the OH Preferences

Probability of Winning.

Remember in the first example our equity changed once we looked at our cards? Well, so too does our TE. This means we need to know our prwin. The more accurate a value for prwin, the more accurately we can asses our TE. We *could* just use the standard OH prwin, but that's not advised. Better to put your opponent (remember we're just up against one person here) on a range of hands and then calculate the prwin. So, how do we do that? Well, there's Weighted Prwin, but I haven't got to grips with that yet. Instead, I went for \$vs\$prwin . We can get our prwin via:

```
vs$listX$prwin + (0.5 * vs$listX$prtie)
```

This is our probabilty of winning (and 1/2 our prop of tying) against listX. LisX# needs to be defined in your .ohf file. For example, say you have put your opponent on the top 5% of hands. Put this in your .ohf

```
##list105##
AA KK QQ JJ TT
AA AKS AQS AJS ATS A9S A8S
AK AQ
Then our prwin will be
vs$list105$prwin + (0.5 * vs$list105$prtie)
If you think your opponent is on any two cards then you can just use
vs$prwin + (0.5 * vs$prtie)
```

So what I did was define hand lists for the top 5,10,20,30,40,60 percent of hands.

I then looked at my opponent's chip stack. If he's got under 5 bb then I reasoned he could be pushing any two cards, 5-10 then pushing the top 50%, over 20BB then pushing the top 10%. Of course this is guess work - some stacks might like to push light with 20BB+, some might push tight. But once you start to look at the figures and change your settings you'll see that in most cases it doesn't make *that* much difference if they're top 20 or top 50 - it can still be a clear push/call whatever.

Probability of our opponent folding/calling.

Just like we need to know the probability of us winning when we enter a pot, we also need to know the chance our opponent will call/fold. For this, I simplified things and again decided my probabilities according to my opponent's chip stack. More than 20BB - 10% call probability, 10-19 - 20%, under 10 - 30%. I hate this oversimplification and can't wait to move to something more reliable (like the Poker Tracker Poker folded blind to steal % maybe).

Putting it all together

So, we have our TEs, we have our prwin and we have the probability that our opponent will fold/call. Now what? Well, it's just like the first example. The pot size was multiplied by our chances of winning it. Here we need to multiply each the TE equity in each "situation" by the chances of that situation happening. Let's take the simple example again: 2 opponents fighting over a \$1 pot, winner takes all, no blinds, 50c stacks.

Do we call a push?

Let's say have AKs again and have worked out our prwin to be 0.65. OH tells us that our icm_callwin is 1.000 and our icm_calllose is 0.000. Simple: we call and win, we get 1. We call and lose: we have 0. We can either:

- Scenario (a.i) Call and win. This probability it will happen is 1 (as we known we're going to call it!) and the prob we win in 0.65. Multiply this by our "icm callwin". This gives 0.65.
- Scenario (a.ii) Call and lose. The prob it happens is again 1, and the prob we lose is (1-0.65). Multiply this by our icm_calllose. This gives 0.

So our total equity for scenario (a) is 0.65.

• Scenario (b) Fold. Prob it happens is 1 if we know we're going to fold. Multiply this with our icm_fold. This gives 0.5.

So, as you can see, scenario (a) gives us more equity. We should call!

Do we push?

Should we push? Same situation as before, but let's add some 1-2 cent blinds and a 20% prob we get called. What are our scenarios?

- Scenario (a.i). Push, get called, and lose. == 1 * (prob of getting called) * (prob of losing) == 1 * 0.20 * (1-0.65) = 0.07. We need to multiply this by our icm_allilose1, which gives us 0.07*0 == 0.
- Scenario (a.ii). Push, get called, and win. = 1 * (prob of getting called) * (prob of winning) == 1 * 0.20 * (0.65) = 0.13. We need to multiply this by our icm alliwin1, which gives is 0.13 * 1.
- Scenario (a.iii). Push, not get called, and win blinds. = 1 * (prob not get called) = 1 * 0.8 = 0.8. We need to multiply this by our icm_alliwin0, which gives us 0.8 * 0.52 (as we win the extra 2 cents from the small blind) = 0.416.

So in total our equity for Scenario a is 0 + 0.13 + 0.416 = 0.546.

• Scenario (b) Fold. Again, our icm_fold which is 0.49 (if we fold, we lose our cent).

So, once again the total equity for scenario (a) beats scenario (b). We should push!

Multiple-Players

Of course, all these calculations are just heads up. Once we go up against multiple players we need much more info. If we're the dealer pushing into the BB and SB, we need to know the likelihood of *both* folding, *both* calling and our prwin against both, or winning against one and not the other and claiming a sidepot. The maths gets a lot more complicated. At the moment I'm just trying to master heads-up, I'm afraid.

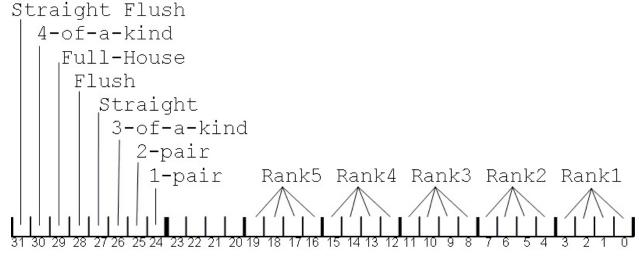
Conclusion

All ICM does is give you a picture of how much your chips are worth in certain scenarios. The real skill is finding out the probability of those scenarios happening. For that, we need accurate prwin and accurate folding probabilities.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-30T22:32:11.031000

Pokerval explained

OpenHoldem classifies and orders poker hands into a single 32bit value.



Bits 31-24 indicate the hand rank

- Straight flush = bit 31 set
- Four of a kind = bit 30 set
- Full house = bit 29 set
- Flush = bit 28 set
- Straight = bit 27 set
- Three of a kind = bit 26 set
- Two pair = bit 25 set
- One pair = bit 24 set
- 5-of-a-kind is indicated when bits 31-24 are set to 1
- High-card (no pair) is indicated when bits 31-24 are set to 0

Bits 23-20 are not used

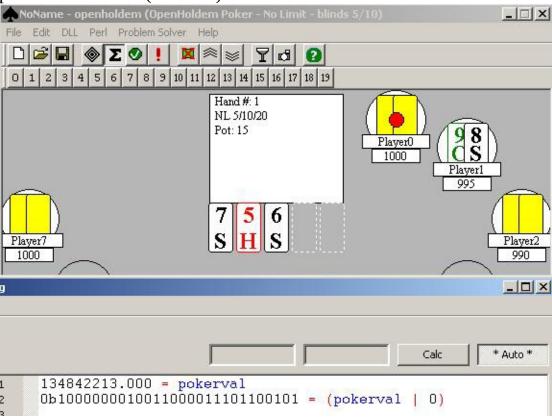
- Bits 19-16 indicate the rank of the top card in the hand
- Bits 15-12 indicate the rank of the second card in the hand
- Bits 11-8 indicate the rank of the third card in the hand
- Bits 7-4 indicate the rank of the fourth card in the hand
- Bits 3-0 indicate the rank of the fifth card in the hand

Numeric values for the ranks (bits 19-1) are:

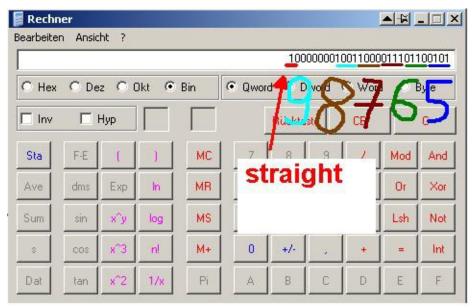
- 14 Ace
- 13 King
- 12 Queen
- 11 Jack
- 10-2 Ten-Two
- 1 Ace (in lo straight)

Pokerval comparisons

The pokerval symbol maps every 0 to 7 card poker hand onto the 32bit number space in correct order of game precedence, allowing two poker hands can be compared to see which is the better hand - higher values are better. The lowest possible value is 0 (no cards).



Now we either apply a binary operation or we feed the pokerval into MicroSofts calculator and then switch to binary mode:



Not that hard to interpret the result.

Player/Pocket Card List

The symbol pcbits is a bit list that indicates where your dealt pocket cards are used in your 5 card poker hand. Ony the lower order 5 bits are significant - one bit per card in your 5 card poker hand. Each bit corresponds to a pokerval rank field as follows:

bit# pokerval bit4 rank5 bit3 rank4 bit2 rank3 bit1 rank2 bit0 rank1. In the example above pcbits would look like

```
pcbits = 00110
```

because we contribute 7 and 6 in a 98765-straight.

If all bits in pcbits are zero then neither of your two cards are used in your 5 card hand. The following formula fragment will be true if you have a sucker straight:

```
((nstraight==5) && (pcbits==1))
```

The following formula fragment will be true if you are filling an inside straight:

```
((nstraight==5) && (pcbits&14))
```

The formula symbol npcbits contains the number of your pocket cards being used in your 5 card poker hand. The possible values are: 0, 1 and 2.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-30T22:46:07.906000

↓Memory Symbols

Introduction

One of the most powerful features of OpenHoldem is its ability to remember. Very few other bots have this ability.

Planning

Working with memory symbols requires a bit of planning and specifically one should keep the following in mind: OpenHoldem does not know your memory symbol before you create it. This sounds simple but one should not forget it because it could create problems with your first hand (The symbol would be undefined and return 0). Once a memory symbol is created, and a value is stored, OpenHoldem will remember that value until it is altered or replaced by another or until OpenHoldem is closed. This is great if you want OpenHoldem to remember what happened 10 hands ago but could bite you if you forget to reset them. It is of utmost importance to decide when and what should be remembered.

Formulation

OpenHoldem provides memory_symbols to store values in variables in your formulas as a replacement for left-hand-side-expressions. Storing a value looks like me_st_SYMBOLNAME_VALUE. For example you can store: numbers, e.g. me_st_Pi_3_141 symbols, e.g. me_st_MyChair_userchair functions, e.g. me_st_CallFunction_f\$call memory symbols, e.g. me_st_OldValue_me_re_CurrentValue me_st_ will store a new value, where_as me_re_ will lookup a previously stored value and me_inc_ will increment a value that serves as a counter. OpenHoldem memory-symbols will never reset as long as you don't explicitly do so. Don't confuse this behaviour with OpenPPL that resets all user-variables at the beginning of a new hand. A technical note: the desired result of a me_st_command is its side-effect; the symbol itself will be evaluated as 0.

Ini-functions

All memory-symbols have to be cleared or initialized at some time. To do this in a clean and fail-safe way OpenHoldem provides six initialization-functions since version 4.2.0. They get called at various times of execution, but always after the basic symbols have been calculated and before the autoplayer-functions get evaluated. Due to our verbosity-fetish they should be pretty self-explaning:

- f\$ini_function_on_startup
- f\$ini_function_on_connection
- f\$ini_function_on_handreset
- f\$ini_function_on_new_round
- f\$ini_function_on_my_turn
- f\$ini_function_on_heartbeat

Simple example:

```
##f$ini_function_on_my_turn##
f$MonsterHand ? me_st_GoodSlowPlaySituation_1 : 0
```

Putting it together

To make things clear we will be putting two memory symbols into your bot. A) We will create a "Stack to Pot Ratio" memory symbol (the function of this symbol is to store the ratio of our balance to the pot at the start of the betting on the flop to be retrieve later in order to make commitment decisions), and B) We will create a "Slow Play" memory symbol, (the function of this symbol is to store our decision during a betting round that we will check-raise the next betting round regardless of what the next card is)

STEP 1: Create the symbols:

In f\$ini_function_on_handreset we put:

```
me_st_stpr_0
+ me_st_slowplay_0
```

to reset these two values on handreset to zero (which means false in the second case).

STEP 2 (a):

```
In f$ini_function_on_my_turn we put
```

```
betround == 2 ?
(me_st_stpr_f$stpr + me_st_slowplay_f$slowplay) : 0
```

Id it is on the flop then we will store the values of the functions f\$stpr and f\$slowplay at that moment.

STEP 2(b)

We create the functions whose value we want to store:

```
##f$stpr##
balance/potcommon
```

Comment: we don't use "pot" because that would include any bets during round2 before us.

```
##f$slowplay##
betround==2 && nhandshi==0 ? 1 : 0
```

Comment: this formula will return 1 when we have the nuts and this will be stored into the "slowplay" memory symbol

STEP 3 Retrieve and use the memory.

In f\$alli we put

```
betround==3 && me_re_slowplay && me_re_stpr<10 ? 1 : 0</pre>
```

English translation: It is the turn and we, on the flop, decided to slow play and the ratio between the pot (at the start of betround==2) and our balance at that time was 1:10 and therefore we will shove.

Conclusion

The memory symbols are very powerful but need a bit of planning. It is important to decide when to define, store and retrieve the values in these symbols.

Rankbits explained

OpenHoldem uses the following method to classify a group of cards according to rank within a single value. Note that if an ace is present then both the 14 bit and the 1 bit will be set.

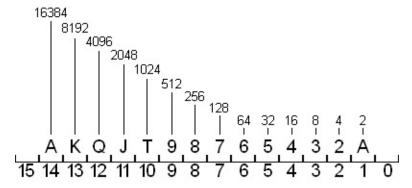
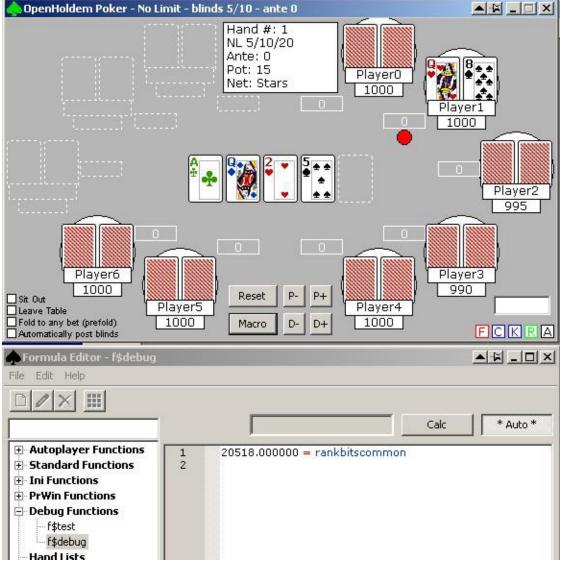


Figure None.1 Rank bits

Bit	Meaning	Decimal value
15	not used	32768
14	Ace (high)	16384
13	King	8192
12	Queen	4096
11	Jack	2048
10	Ten	1024
9	9	512
8	8	256
7	7	126
6	6	64
5	5	32
4	4	16
3	3	8
2	2	4
1	Ace (low)	2
0	not used	1



In this example rankbits is 20518 (decimal), which is 0b101000000100110 (binary)

```
101000000100110
A-Q----5--2A-
```

A quick calculation will verify that 20518 is the sum of the card-values in the table above:

Ace (high)	16384
Queen	4096
Five	32
Two	4
Ace (low)	2

rankbits (sum) 20518

Now a quick check, if an ace is on the board:

```
##f$ace_on_the_board##
(rankbitscommon & 0x4000) != 0
```

This code will just execute a bit-wise AND-operation of the ACE-high-bit and then check if the result is non-zero, i.e ace present. 0x4000 is the hexadecimal representation of 0b0010000000000000, that's just more easy to write and read than plain binary, and very easy to transform.

If you have never worked with binary numbers before — don't get confused. Just try out MicroSofts calculator and select scientif view. Things will become easy.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-30T22:33:30.921000

Handstrength-library

In the legendary golden age of Fixed Limit Hold'em Winngy, member of Lab 1, developed a hand-strength-library that rates all hands from 1 (weak) to 5 (very strong). He used it successfully in his private Fixed-Limit-bot that played \$5/\$10 at that time. For the No-Limit-guys of modern times: these were the blinds, not buyins; and he also got a 100%-prop-deal to make his hobby even more enjoyable. Long story short: Fixed Limit died slowly and so Winngy decided to donate his hand-ranking-system to the OpenHoldem project. Our member HTC added his outs-counter. Both are combined in a library, which can be downloaded from

https://github.com/OpenHoldem/openholdembot/blob/master/mh_str_Handstreng make use of these functions just include this library in your bot-logic.

	Meaning
0 4 1 f 1 0 4)	(1 f d1 0+1 f b d+)
,	(x=1 for wheel, 0 not, y=1 for broadway, 0 not) - returns true if the board has a wheel straight draw
•	or broadway straight draw, given the
	wheel/broadway parameters 00 - true if the board
1	has neither a wheel straight draw nor a broadway
	straight draw 10 - true if the board has a wheel
	straight draw 01 - true if the board has a broadway
	straight draw 11 - true if the board has either a
·	wheel straight draw or a broadway straight draw
• • •	S \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
5	true e.g. hole: T2 common: 345K)
ng suited cards that are higher than	number of missing suited cards that are higher than
ard	my best suited card
<u> </u>	number of cards that can beat your kicker
· ·	` `
,	
`	·
ne relative strength of your straight	flush
	0-5 (5 best) of the relative strength of your four of a
oottom straight draw (if you are ingle card to an open-ended straight ard is the smallest, this symbol is 2 common: 345K) and suited cards that are higher than eard that can beat your kicker k rank of your kicker (returns 0 if [board] and thus useless) (HTC's formula) f\$mh_str_strflush	true if I have a bottom straight draw (if you are contributing a single card to an open-ended straight draw and that card is the smallest, this symbol is true e.g. hole: T2 common: 345K) number of missing suited cards that are higher than my best suited card

	kind
f\$mh_str_fullhouse	0-5 (5 best) of the relative strength of your full house
f\$mh_str_flush	0-5 (5 best) of the relative strength of your flush
f\$mh_str_straight	0-5 (5 best) of the relative strength of your straight
f\$mh_str_trips	0-5 (5 best) of the relative strength of your three of a kind
f\$mh_str_twopair	0-5 (5 best) of the relative strength of your two pair
f\$mh_str_onepair	0-5 (5 best) of the relative strength of your one pair

MyHand explained

This chapter contains a detailed explanation of the Winngys f\$mh_-symbols, written by SNGChamp for the Wiki. Basically, the f\$mh_str-symbols (e.g. f\$mh_str_twopair) return a relative ranking of your hand (hole and common) from 1 (weak) to 5 (strong), but they can be discounted up to 3 points depending on the board-texture. The handstrength symbols are a separate library in text-format and can be downloaded from TODO!!!!!

f\$mh_str_onepair

- 1 = Board has Pair and you have Two small cards
- 1 = Board has Pair and you have "One overcard w/ King (K9-K2)"
- 1 = Board has Pair and you have "One overcard w/Ace (A9-A2)"
- 1 = Board has Pair and you have Two overcards
- 1 = Board has No Pair and you have Bottom pair w/ poor kicker
- 1 = Board has Pair and you have "Two overcards w/ King"
- 1 = Board has Pair and you have "Two overcards w/Ace"
- 2 = Board has No Pair and you have Middle pair w/ poor kicker
- 2 = Board has No Pair and you have Underpair
- 2 = Board has No Pair and you have Bottom pair w/ overcard kicker
- 2 = Board has No Pair and you have Midpair 2 overcard
- 3 = Board has No Pair and you have Middle pair w/ overcard kicker
- 3 = Board has No Pair and you have Top pair Queen or lower w/ poor kicker (9-2)
- 3 = Board has No Pair and you have Top pair King w/ poor kicker (9-2)
- 3 = Board has No Pair and you have Midpair 1 overcard
- 4 = Board has No Pair and you have Top pair Ace w/ poor kicker (9-2)
- 4 = Board has No Pair and you have Top pair Queen or lower w/ good kicker (A-T)
- 4 = Board has No Pair and you have Overpair
- 5 = Board has No Pair and you have Top pair King w/ good kicker (A-T)
- 5 = Board has No Pair and you have Top pair Ace w/ good kicker (K-T)
- 5 = Board has No Pair and you have Overpair 10's or better

OnePairDiscount

If the board shows 3 or 4 suited cards OR if the board shows 4 connecting cards, the strength value is reduced by 1.

f\$mh_str_twopair

- 1 = Board has Pair and you have Two pairs w/ poor kicker (9-2)
- 1 = Board has two pair and you have only two pair
- 2 = Board has Pair and you have Two pairs w/ good kicker (A-T)
- 2 = Board has Pair and you have Underpair
- 3 = Board has No Pair and you have Bottom two pairs
- 3 = Board has Pair and you have Overpair
- 4 = Two pair with a Rainbow board
- 4 = Board has No Pair and you have Top and bottom pair
- 5 = Board has No Pair and you have Top two pairs

TwoPairDiscount

If the board shows 3 or 4 suited cards OR if the board shows 4 connecting cards, the strength value is reduced by 1.

f\$mh_str_trips

- 1 = Board has trips
- 2 = Trips using 1 of your hole cards (w/ paired board)
- 2 = Board has trips but you have good kickers (10 to A)
- 2 = Board has Pair and you have Trips w/ poor kicker (not used)
- 3 = Board not paired catch all
- 3 = Board has No Pair and you have bottom set
- 3 = Board has Pair and you have Trips w/ moderate kicker (used as 5th ranked card)
- 4 = Board has Pair and you have Trips w/ good kicker (used as 4th ranked card)
- 4 = Board has No Pair and you have middle set
- 5 = Board has No Pair and you have top set

TripsDiscount

If the board shows 3 or 4 suited cards OR if the board shows 4 connecting cards,

the strength value is reduced by 1.

f\$mh_str_straight

- 1 = Board has straight
- 1 = Board 4 to a straight, card used low end of straight (rank 5)
- 2 = Board 4 to a straight, card used in rank 4
- 2 = Board 4 to a straight, card used in rank 3
- 3 = Board 4 to a straight, card used in rank 2
- 3 = Board 4 to a straight, card used in rank 1
- 3 = Board 3 to a straight, cards in rank 4 and rank 5 (sucker's straight)
- 3 = Board 3 to a straight, cards in rank 3 and rank 5
- 3 = Board 3 to a straight, cards in rank 3 and rank 4
- 3 = Board 3 to a straight, cards in rank 2 and rank 5
- 4 = Board 3 to a straight, cards in rank 2 and rank 4
- 4 = Board 3 to a straight, cards in rank 2 and rank 3
- 4 = Board 3 to a straight, cards in rank 1 and rank 5
- 5 = Board 3 to a straight, nut straights
- 5 = Board 4 to a straight, card used in rank 5, nut straight with a Ten
- 5 = Board 4 to a straight, card used in rank 4, nut straight with a Jack
- 5 = Board 4 to a straight, card used in rank 3, nut straight with a Queen
- 5 = Board 4 to a straight, card used in rank 2, nut straight with a King
- 5 = Board 4 to a straight, card used in rank 1, nut straight with a Ace
- 5 = Board 3 to a straight, cards in rank 1 and rank 4
- 5 = Board 3 to a straight, cards in rank 1 and rank 3
- 5 = Board 3 to a straight, cards in rank 1 and rank 2 (high end)

StraightDiscount

- If the board shows at least 3 suited cards, the strength value is reduced by 1.
- If the board shows at least 4 suited cards, the strength value is reduced by 2.
- If the board shows trips, the strength is reduced by 1.
- If the board shows 2 pair, the strength is reduced by 2.
- StraightDiscount is negated if there is a good draw (+1)

f\$mh_str_flush

• 1 = board has flush, my cards don't help

- 1 = Board has trips or more
- 1 = Four or more cards can beat my flush
- 2 = Three or less cards can beat my flush
- 3 = Three or more cards can beat my flush, board may be paired, flush came on turn [something doesn't seem right here...]
- 3 = Two or more cards can beat my flush, board not paired
- 4 = One card can beat my flush
- 5 = Nut flush

FlushDiscount

If the board shows two pair, the strength value is reduced by 2.

f\$mh_str_fullhouse

- 1 = Board has full house
- 2 = Board trips, only one of my hole cards participates, the pair rank is lower than non-participating board cards e.g. QT/999TK
- 2 = Board trips, only one of my hole cards participates, the pair rank is higher than non-participating board cards e.g. QT/999T8
- 2 = Board trips, I have pocket pair lower than non-participating board cards e.g. QQ/JJJK2
- 2 = Board trips, I have pocket pair higher than non-participating board cards e.g. QQ/JJJ92
- 3 = none
- 4 = Board has two pair, I have top trips e.g. QT/QQJJ8
- 4 = Board paired, I don't have pocket pair, I have bottom side full house, pair is lower than non-participating board cards e.g. QJ/QJJ53
- 4 = Board paired, I don't have pocket pair, I have bottom side full house, pair is higher than non-participating board cards e.g. QJ/QJJ53
- 4 = Board paired, I don't have pocket pair, I have top side full house, pair is lower than non-participating board cards e.g. QJ/QQJA3
- 4 = Board paired, I don't have pocket pair, I have top side full house, pair is higher than non-participating board cards e.g. QJ/QQJ53
- 5 = Board paired, I have pocket pair and bottom side full house, pocket pair is lower than non-participating board cards e.g. QQ/QKKA6
- 5 = Board paired, I have pocket pair and bottom side full house, pocket pair is higher than non-participating board cards e.g. QQ/QKKT6

- 5 = Board paired, I have pocket pair and top side full house, pocket pair is lower than non-participating board cards e.g. QQ/QTTA3
- 5 = Best Full House; Board paired, I have pocket pair and top side full house, pocket pair is higher than non-participating board cards e.g. QQ/QTT53

f\$mh_str_quads

- 1 = Board has quads, I don't have best kicker
- 2 = Board has quads, I have a King kicker
- 3 = I may not have the highest, board does not have quads
- 4 = none
- 5 = Board quads, I have an Ace
- 5 = Nut 4 of a kind

f\$mh_str_strflush

- 1 = Chop pot
- 2 = I may not have the highest, board does have a str8flush
- 3 = I may not have the highest, board does not have a str8flush
- 4 = none
- 5 = nuts

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-30T21:26:21.781000

DLL

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:18:31.593000

User-DLL

OpenHoldem has the ability to interface with a user-compiled DLL. Microsoft Windows provides a feature whereby running code can load other code, presuming that code is provided in the form of DLL files. OpenHoldem utilizes this Windows DLL feature to allow you to extend the functions within OpenHoldem to practically anything that you want, provided you can write the code for it and compile it. Generally, people who create DLLs write them in the C or C++, as those are the native historical target languages for using DLL functions, as defined by Microsoft. There are ways to have OpenHoldem use a C/C++ DLL shell which then calls code written in other languages, such as managed languages like .NET, but that topic is beyond the scope of this manual. If you are interested in doing this, there are some topics in our C++ subforum that talk about utilizing other languages.

OpenHoldem provides a number of preferences for how a user-DLL will be loaded. It can be done on start-up, when a formula is loaded, or manually. See the {Preferences} section for more information.

From this point on, the assumption is that, if you are creating a user-DLL, it will be written in the C or C++ languages. To do this, you need a C/C++ compiler — the express editions of Microsoft Visual Studio will suffice for this, as these user-DLLs typically do not need the MFC library, which is only present in the paid versions of Microsoft Visual Studio. A starting framework for a user-DLL can be found in the source code library on Google Code, but is also embedded just below this paragraph. See the {References} section for links.

How OpenHoldem and the DLL cooperate

The OpenHoldem to DLL interface is small and clean. OpenHoldem exports the following functions and the DLL gets linked to them at load-time.

```
// Functions exported by OpenHoldem
extern "C" __declspec(dllimport) double __stdcall GetSymbolFromDll(c
extern "C" __declspec(dllimport) void __stdcall SendChatMessageFomDl
extern "C" __declspec(dllimport) void* __stdcall GetPhl1kFromDll();
extern "C" __declspec(dllimport) void* __stdcall GetPrw1326FromDll()
extern "C" __declspec(dllimport) void __stdcall WriteLogFromDll(char
```

And the DLL will export a function too, namely

```
__declspec(dllexport) double process_message( const char* message, c
```

Again linking happens at load-time, so the DLL-interface is ready for cooperation right from the start. Contrary to WinHoldem and early versions of OpenHoldem no function-pointer get exchanged at run-time, so a very error-prone mechanism could get removed.

OpenHoldem will send or request information from the DLL when needed, and it is up to the DLL's author to process these requests. The exported library function process_message is used for this interface and has the following prototype:

```
USERDLL_API double process_message (const char* pmessage, const void
```

The various messages that can be passed in the "pmessage" parameter are as follows: pmessage meaning and parameters state

Game-States

Each scrape cycle, OpenHoldem will send the current game state to the DLL. As the author of the DLL, you will decide what to do with this game state information. The game state information is stored in a structure within OpenHoldem, and a pointer to this structure is passed to your User DLL in the "param" parameter. An example of how to cast this void pointer to the game state structure is provided in the Reference User DLL source code. The details of what is included in this game state structure are in the header file in the Reference User DLL source code.

Querying dll\$-Symbols

Any symbol that begins with the characters "dll\$" is interpreted by the OpenHoldem parser as requiring a call to the loaded User DLL in order to determine its value. The "query" message is the mechanism for requesting the value of this symbol. Any symbol name can be created for use by your User DLL, as long as it begins with "dll\$", and your DLL code knows how to handle it. So, for example, if the OpenHoldem decision engine encounters a symbol called "dll\$kill_phil" in the evaluation of the "f\$alli" function, then a synchronous call will be made to the User DLL to provide the value of dll\$kill_phil in order to complete the evaluation of f\$alli. The evaluation of f\$alli will block until your DLL handles this request and provides a result. The name of the symbol (in this case, "dll\$kill_phil") is provided in the "param" parameter, and should be cast to a (const char*). A simple example of how to process this message is provided in the Reference User DLL source code.

load and unload Messages

There are two event messages that are passed to your DLL. Both of these messages have NULL "param" parameters. "load" – this message is sent once immediately after your DLL is loaded. Typically this is used for some kind of initialization activity by your DLL. "unload" – this message is sent once just prior to OpenHoldem unloading your DLL from memory. This could be due to OpenHoldem shutdown, manual intervention from the DLL menu, or new formula loading. Typically this is used to clean up your DLL's activities, such as connections to databases, closure of threads, etc.

phl1k

This is a pointer to OpenHoldem's internal hand list array, for your DLL to read and modify as needed. It points to an array of the following type:

```
bool inlist[1000][13][13];
```

OpenHoldem can store 1000 different hand lists, and the first dimension of this array specifies the hand list number, the second and third dimensions specify the rank of the first and second card, where rank0>=rank1 is interpreted to mean suited cards, and rank0<rank1 is interpreted to mean unsuited cards. This is a very sparsely populated array, where the intersection of the three dimensions

specifies if the particular card pair is present in the hand list or not. A non-zero value at this intersection indicates that the card pair is present; zero indicates that the card pari is not present. prw1326 This message provides a pointer to a prw1326 structure stored in OpenHoldem. This prw1326 structure is used to describe the manner in which OpenHoldem calculates its prwin, prtie and prlos symbols, based on the information which the DLL developer places in this structure. Unless the DLL code sets a specific enabling flag in this structure, the current prwin, prtie and prlos calculation is unchanged. Please see the {prw1326} section for more information.

OpenHoldems handlist-structure got rewritten and OpenHoldem is no longer restricted to 1000 numbered lists. Therefore this paragraph does currently not apply. A new machanism for manipulating handlists will be developed in the near future.

A Simple Example

An easy way to get started is to override all OH-Script processing with DLL functions. There are a number of reasons to do this, including:

- Limitations of the OH-Script language (no assignments is the biggest limitation)
- Low level integration with other technologies

To override all OH-Script processing, simply instruct each primary function to only have a call to a "dll\$" symbol. For example, use the following code:

```
##f$alli##
dll$alli
##f$rais##
dll$rais
```

And so on. In your User DLL, handle each of these "dll\$" symbols as described above. Now all of the decision logic has been moved from OH-Script code to code that you provide in your DLL. It is important to remember that even if you want to code up most of your bot logic in C/C++, you still must use the OpenHoldem primary functions themselves to call this logic.

prw1326

The prw1326 structure and associated individual chair structure have the following specification:

```
//prwin 1326 chair structure
struct sprw1326_chair
    int level; // indicates weighting level
                // for 'always consider'
               // max index into weight array
    int limit;
                // used for computational efficiency
    int ignore; // if non-zero no weighting will
                // be applied to this chair
    int rankhi[1326]; // higher ranked card number
                      // in pocket cards
    int ranklo[1326]; // lower ranked card number
                      // in pocket cards
    int weight[1326]; // the significance value for this hand
double scratch: // for future reference
};
//prwin 1326 structure
struct sprw1326
{
    int useme; // unless set to 1326 the normal
               // OH prwin will be used
    int preflop; // unless set to 1326 the normal
                 // OH prwin will be used pre-flop
    int usecallback; // unless set to 1326 the callback
                     // function will not be invoked
    double (*prw callback)(void); // if enabled will be
                                  // invoked before the
                                  // prwin calculation pass
    double scratch; // for future reference
    int bblimp; // if non-zero no weighting
                // will be applied if a chair has put
                // nothing in the pot
    sprw1326_chair vanilla_chair; // will be precalculated
                                  // by OH at startup.
                                  // convenience values
    sprw1326_chair chair[10]; // structures for each chair
};
To use the structure from a User DLL, the following would also need to be
added:
typedef sprw1326* pp13; pp13 prw1326;
Also, in the process_message handling add:
```

During OpenHoldem initialization the prw1326.vanilla_chair structure is set up so that .level is 1024, the .ranklo and .rankhi arrays contain the card order used for the normal OpenHoldem prwin calculation (but expanded from 169 to 1326 format), the weighting of the high third of these cards is set to 1024, the middle third is sloped down from 1024 to zero, and .limit is set so that the final third are never even considered as possible opponent cards. The .vanilla_chair is then copied to each of the ten .chair structures. This should mean that even if the capability is activated by setting .useme to 1326, the prwin calculation will not crash and will give an anodyne weighted result. The vanilla_chair is not referenced or used by OpenHoldem other than at start-up, and could be altered and used by the DLL developer to act as a default profile for new players. A more complete explanation of how the prw1326 approach works can be found in the {Weighted prwin} section.

isfinalanswer

isfinalanswer is a calculated symbol, designed for User-DLL developers, who want to make sure, that certain calculations get executed only once per turn. This symbols is true (non-zero) just prior to the Autoplayer acting, and false (zero) any other time. More precise, it is true exactly when

- it is your turn
- you see stable frames, i.e. when N scrapes is a row are identical (N is set in Edit->Preferences->Autoplayer->Frame Delay)

dll\$iswait

Prior to the OpenHoldem Autoplayer taking action, it will query your User DLL, if one is loaded, for the value of dll\$iswait. If this query returns non-zero, then OpenHoldem will not execute the Autoplayer's action this scrape cycle, but rather will skip the action and continue to the next screen scrape cycle. While dll\$iswait returns non-zero, isfinalanswer will always be zero. In the forum Matrix explained

(http://www.maxinmontreal.com/forums/viewtopic.php?f=174&t=6885):

While dll\$iwait is true you don't have to respond to formula requests. You do have to somehow make sure that by the time dll\$iswait is true that

OpenHoldem has been returned the correct result. When you go down this route you may encounter the problem of OpenHoldem stop evaluating the formulas, because there has been no change in state and dll\$iswait is false. The best thing to do in this case is to disable all caching inside of OpenHoldem and always return a valid formula result and set dll\$iswait to false on the very first valid frame with buttons you see from OpenHoldem.

In short: dll\$iswait will be only needed if you do very time-consuming computations in your DLL that get executed in another thread. Just return true to tell OH: "Hey buddy, I am really busy atm. Please come back later and do something else in the meantime. Scrape some additional frames, sitin/sitout/rebuy etc." Most DLLs should just return false all the time. isfinalanswer: an OH-symbol that becomes true, when all conditions are met that are required that the autoplayer acts (enough stable frames, dll\$iswait = false, balance stability, etc.) If you want to execute your DLL-logic only once, you should wait for isfinal-answer = true and then calculate it on the fly. OH will be blocked for a small moment while your DLL is working.

cmd\$recalc

A User DLL also has the ability to send the request "cmd\$recalc", which will force OpenHoldem to recompute the versus tables and restart the Iterator thread.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-28T22:25:51.375000

Compiling the first dll

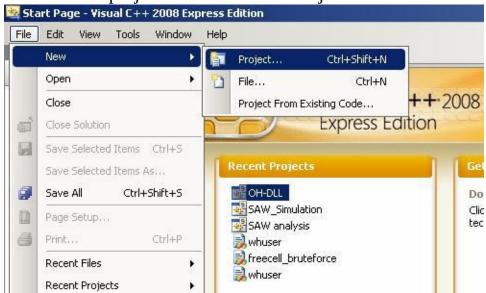
This tutorial got written by VaDe for the Wiki (http://www.maxinmontreal.com/wiki/index.php5? title=Part A: Compiling the first dll).

Step 1

Download MSVSE 2008 here: http://www.microsoft.com/express/download/ - Install with custom settings; there is no need to change anything.

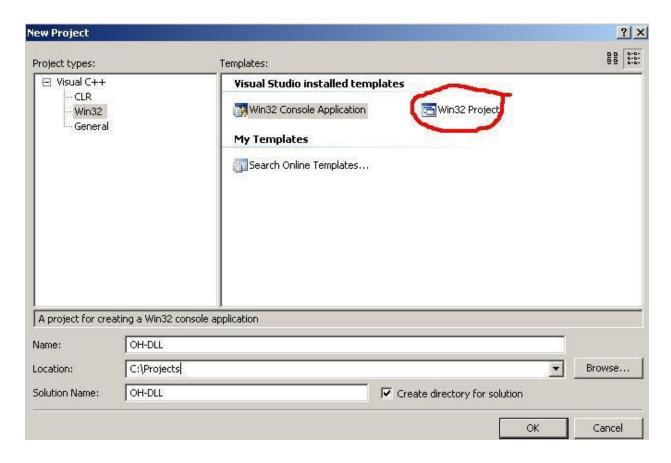
Step 2

Create a new project: File -> New -> Project.

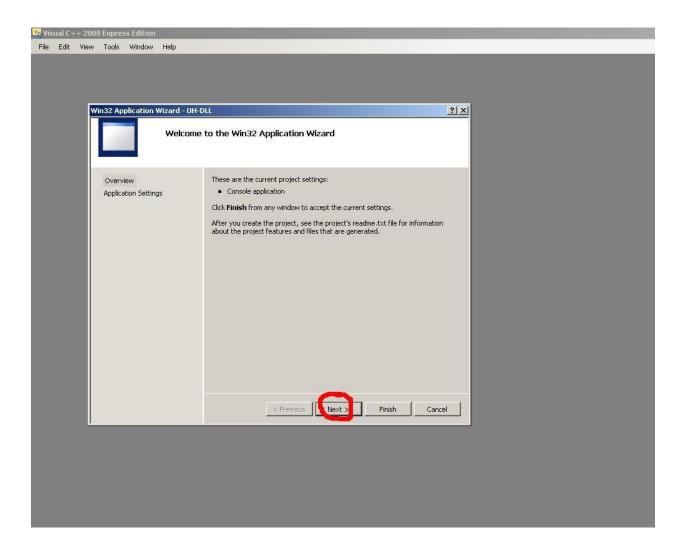


Step 3

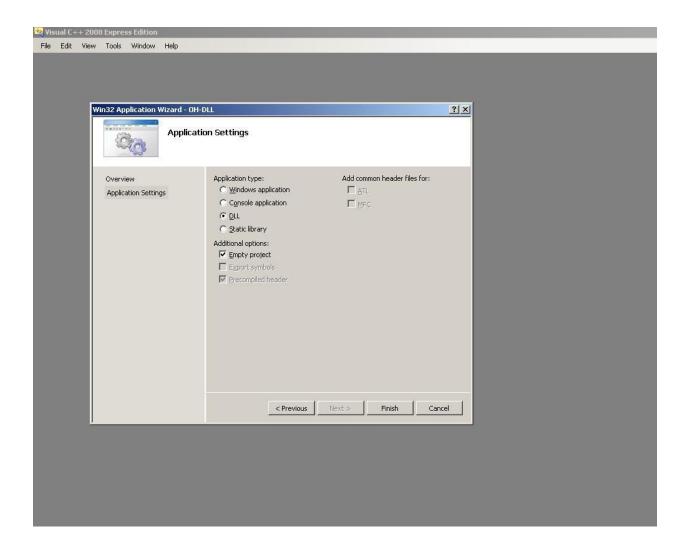
Choose Win32 Project Application, you can choose any name/directory you want for your project, but I suggest you stick with OH-DLL and C:\Projects for this tutorial



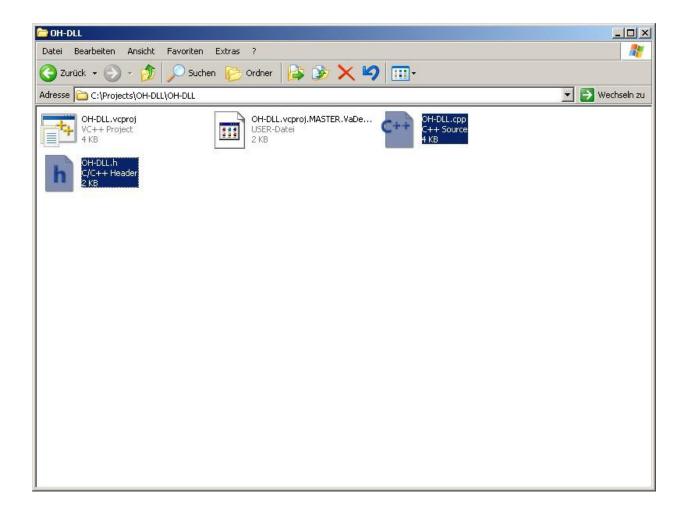
Don't click the "Finish" button yet, but click on "Next".



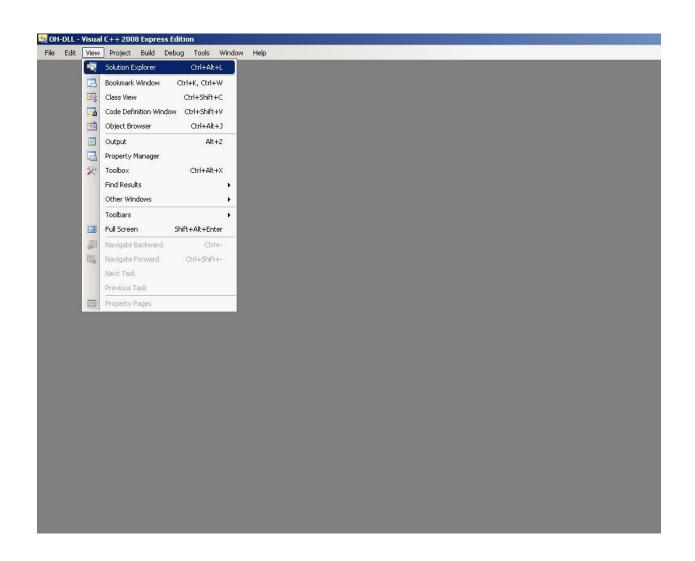
Select "DLL" and "Empty project" on the following screen and click the "Finish" button now.

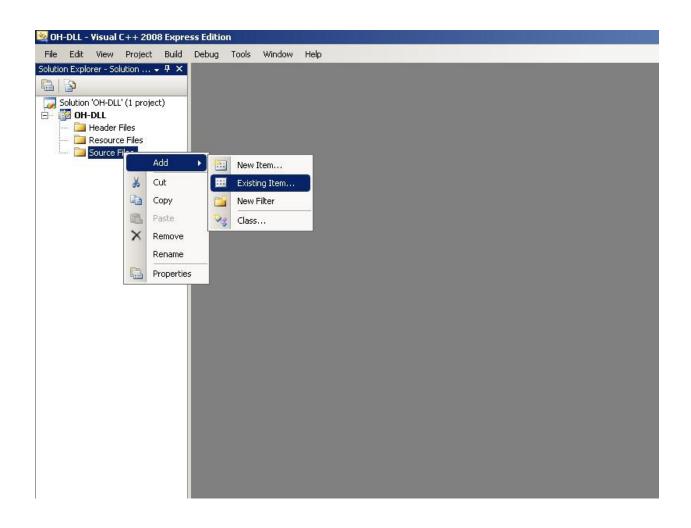


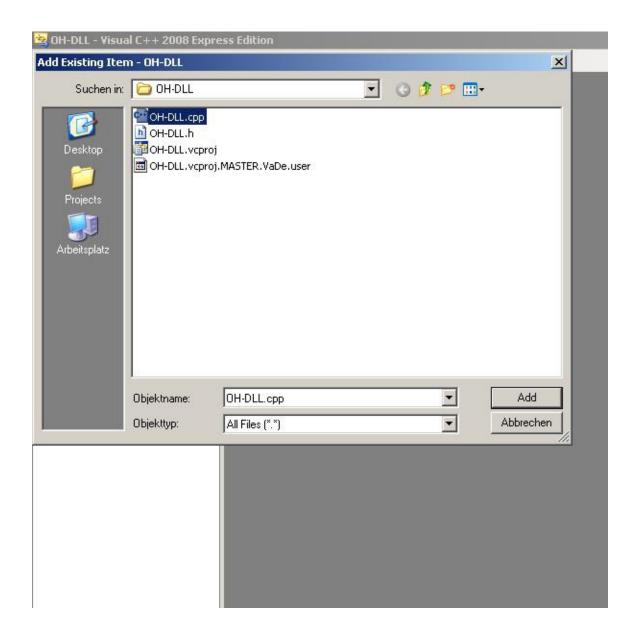
Download and copy the attached files into the project folder, as you can see on the screen it should be C:\Projects\OH-DLL\OH-DLL. The files are available at the Wiki: http://www.maxinmontreal.com/wiki/index.php5?title=Image:OH-DLL.rar.



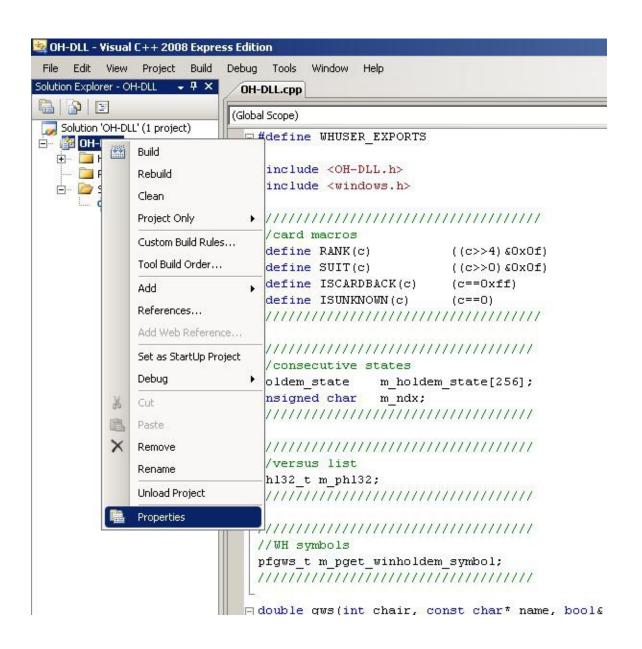
- Open the Solution Explorer: View -> Solution Explorer
- Add the files to the project, select the "Source File" folder in the Solution Explorer -> Add -> Existing Item
- Mark "OH-DLL.cpp" in your project folder and click add
- Select the "Header File" folder and add the header file the same way

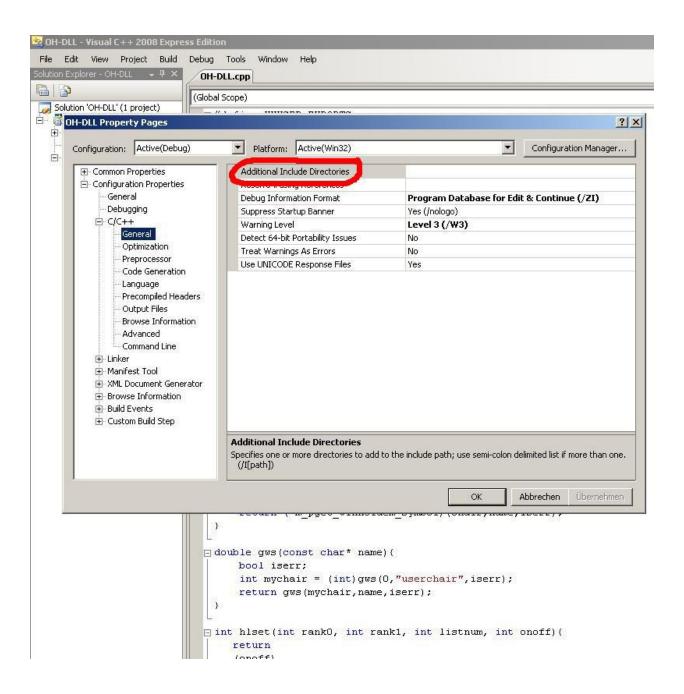


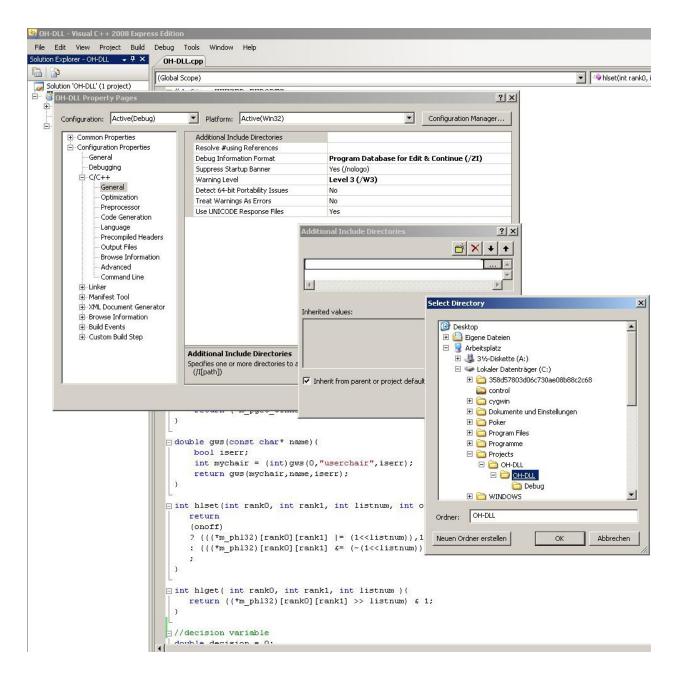




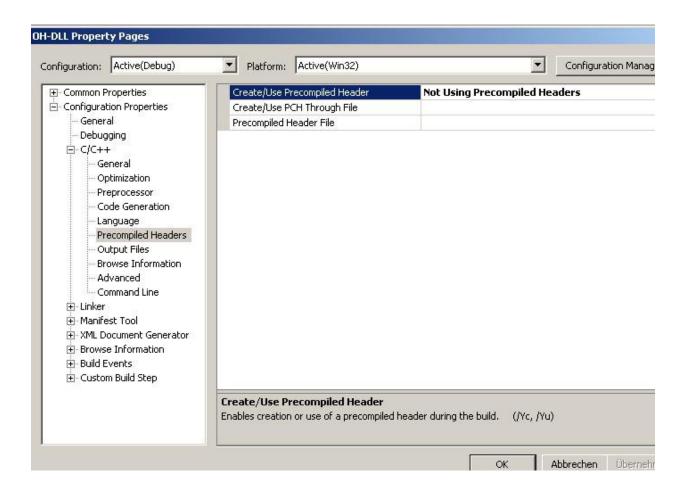
- Now we need to change some options, right-click on the main tree OH-DLL, select Properties
- Open the "C/C++" tree, select "General", select Additional Include Directories
- Add the project folder here, c:\projects\oh-dll\oh-dll



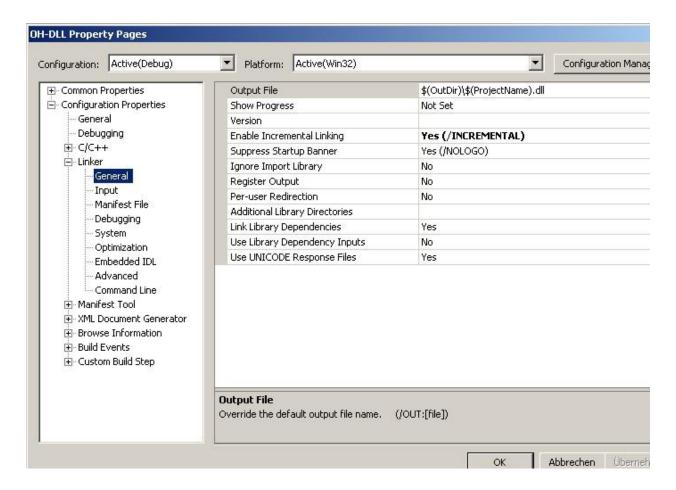




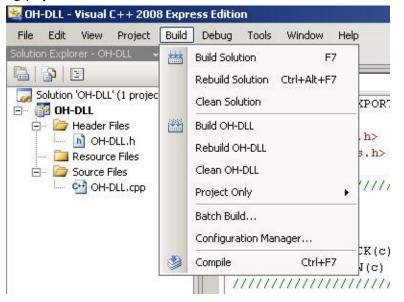
Select "Precompiled Headers" in the "C/C++" tree and select "Not Using Precompiled Headers".



In the "Linker" tree under "General" you can change the destination of the dll Output File if you need.



You are ready to compile the dll, click on Build -> Build Solution or simply "F7".



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-28T22:30:13.531000

Basic overview over the dll structure & examples

If you are not following my guide "Compiling the first dll" there might be slight differences in the basic files you use, since i already added some code, which is needed anyway.

The OH-DLL.cpp

```
#define WHUSER_EXPORTS
#include "OH-DLL.h" #include <windows.h>
//card macros
#define RANK(c)
                ((c>>4)\&0x0f)
#define SUIT(c)
                  ((c>>0)&0x0f)
#define ISCARDBACK(c)
                  (c==0xff)
#define ISUNKNOWN(c)
                  (c==0)
//consecutive states
holdem_state
           m_holdem_state[256];
unsigned char
           m ndx;
//versus list & prwin
phl1k_t m_phl1k;
pp13 prw1326;
//WH symbols pfgws t m pget winholdem symbol;
double gws(int chair, const char* name, bool& iserr) {
 return (*m pget winholdem symbol)(chair, name, iserr);
}
double gws(const char* name) {
 bool iserr;
 int mychair = (int)gws(0, "userchair", iserr);
 return gws(mychair, name, iserr);
}
bool hlset( int rank0, int rank1, int listnum, bool onoff ) {
 return ((*m_phl1k)[listnum][rank0-2][rank1-2] = onoff);
}
bool hlget( int rank0, int rank1, int listnum ) {
```

```
return ((*m_phl1k)[listnum][rank0-2][rank1-2]);
}
double process_query(const char* pquery){
 if(pquery==NULL)
   return 0;
             //example
 if(strcmp(pquery, "dll$test")==0)
   return 101;
 return 0;
double process_state(holdem_state* pstate){
 if(pstate!=NULL)
   m holdem state[ (++m ndx)&0xff ] = *pstate;
 return 0;
}
//WINHOLDEM RUNTIME ENTRY POINT
WHUSER_API double process_message(const char* pmessage, const void*
 if(pmessage==NULL){
   return 0;
 if(param==NULL)
  return 0;
 if(strcmp(pmessage, "state")==0)
       return process_state( (holdem_state*)param );
 if(strcmp(pmessage,"phl1k")==0) {
   m_{phl1k} = (phl1k_t)param;
   return 0;
 if(strcmp(pmessage, "prw1326")==0) {
   prw1326 = (pp13)param;
   return 0;
 if(strcmp(pmessage, "query")==0)
   return process_query((const char*)param);
 if(strcmp(pmessage, "pfgws")==0) {
   m_pget_winholdem_symbol = (pfgws_t)param;
   return 0;
 return 0;
}
//DLLMAIN
BOOL APIENTRY DllMain( HANDLE hModule, DWORD ul_reason_for_call, LPV
 switch(ul_reason_for_call) {
   case DLL PROCESS ATTACH: break;
```

```
case DLL_THREAD_ATTACH: break;
  case DLL_THREAD_DETACH: break;
  case DLL_PROCESS_DETACH: break;
}
return TRUE;
}
```

gws function

```
double gws(const char* name) {
  bool iserr;
  int mychair = (int)gws(0,"userchair",iserr);
  return gws(mychair,name,iserr);
}
```

This function you see here, is very important, because you can access all symbols, which are available for Openholdem with it. E.g. gws("userchair") returns the value of the userchair symbol.

hlset & hlget

```
bool hlset( int rank0, int rank1, int listnum, bool onoff ) {
  return ((*m_phl1k)[listnum][rank0-2][rank1-2] = onoff);
}
bool hlget( int rank0, int rank1, int listnum ) {
  return ((*m_phl1k)[listnum][rank0-2][rank1-2]);
}
```

These functions can be used to access and dynamically modify versus lists during play e.g:

```
hlset( 14, 13, 5, 1);
```

this sets AKs on list 5 to true. More details can be found here: http://www.maxinmontreal.com/wiki/index.php5?title=Phl1k.!!!

process_query function

Inside this scope you can define symbols which you can access with the OH formula editor.

E.g: If you add " = dll\$test" in the OH formula editor in the debug tab, you'll get the value "101". You can experiment with it if you want and try some values from the gws function, but remember to recompile each time you change something in the code.

process_state function

```
double process_state(holdem_state* pstate){
  if(pstate!=NULL) m_holdem_state[ (++m_ndx)&0xff ] = *pstate;
  return 0;
}
```

The OH-DLL.h Header

```
#ifndef _whuser_h_ #define _whuser_h_
#ifdef WHUSER_EXPORTS #define WHUSER_API __declspec(dllexport)
#else
#define WHUSER API declspec(dllimport) #
endif
struct holdem_player{
                m_name[16];
                             //player name if known
 char
                m_balance; //player balance
 double
                m_currentbet; //player current bet
 double
                m_cards[2] ; //player cards
 unsigned char
 unsigned char
                m_name_known : 1 ; //0=no 1=yes
 unsigned char
                m_balance_known : 1 ; //0=no 1=yes
                m_fillerbits : 6 ; //filler bits
 unsigned char
 unsigned char
                m_fillerbyte
                              ; //filler bytes
};
struct holdem_state{
                char
 double
 unsigned char
 unsigned char
 unsigned char
 unsigned char
                               ; //filler byte
 unsigned char
                m_dealer_chair ; //0-9
m_player[10] ; //play
 unsigned char
 holdem_player
                                  ; //player records
};
struct sprw1326_chair{
```

```
level;
                       //indicates weighting level for 'always consi
  int
        limit;
                       //max index into weight array - used for comp
  int
        ignored;
  int
                       //if non-zero no weighting will be applied to
                       //higher ranked card number in pocket cards
  int
        rankhi[1326];
        ranklo[1326];
  int
                      //lower ranked card number in pocket cards
  int
        weight[1326];
                      //the significance value for this hand
                           //for future reference
  double scratch;
};
struct sprw1326{
                      //unless set to 1326 the normal OH prwin will
  int
        useme;
  int
                     //unless set to 1326 the normal OH prwin will
        preflop;
        usecallback; //unless set to 1326 the callback function wil
  int
  double (*prw_callback)(void); //if enabled will be invoked before
  pass
  double scratch;
                         //for future reference
  //if non-zero no weighting will be applied if a chair has put noth
  //will be precalculated by OH at startup - convenience values
  sprw1326_chair vanilla_chair;
  sprw1326_chair_chair[10]; //structures for each chair
};
typedef double (*process_message_t)(const char* message, const void*
WHUSER_API double process_message( const char* message, const void*
typedef double (*pfgws_t)( int c, const char* psym, bool& iserr );
typedef bool hl1k_t[1000][13][13]; // list number, rank0, rank1
// rank0>=rank1 == suited, rank0<rank1 == unsuited</pre>
typedef hl1k_t* phl1k_t;
typedef sprw1326* pp13;
#endif
```

Here you have some definitions of datatypes. The important definitions are the two structs, you can basically compare a "struct"-datatype to an object with some attributes, these attributes are accessed by using the "." operator, look in the examples section below.

The struct holdem_state

```
struct holdem state{
  char m_title[64]
                                   //table title
                   m_pot[10]
  double
                                         ; //total in each pot
                                          ; //common cards
  unsigned char
                   m_cards[5]
                   m_is_playing : 1 ; //0=sitting-out, 1=sitting-i
m_is_posting : 1 ; //0=autopost-off, 1=autopost
  unsigned char
  unsigned char
  unsigned char
                   m_fillerbits
                                     : 6 ; //filler bits
                                         ; //filler byte
  unsigned char
                   m_fillerbyte
                                         ; //0-9
  unsigned char
                   m dealer chair
                   m_player[10]
  holdem player
                                          ; //player records
```

- What you basically have here is the information of a OH state which was scraped.
- You can access information like the tabletitle/potsize/dealerchair etc. Ofcourse that is nothing new, as you have symbols formost of that at the formula level.
- It also holds information about the players which is saved in the "struct" "holdem_player".

The struct holdem_player

```
struct holdem_player{
                                           ; //player name if known
                     m_name[16] ; //player name if know
m_balance ; //player balance
m_currentbet ; //player current bet
m_cards[2] ; //player cards
  char
  double
  double
  unsigned char
                     m_name_known : 1 ; //0=no 1=yes
  unsigned char
  unsigned char
                     m_balance_known : 1 ; //0=no 1=yes
  unsigned char
                     m_fillerbits : 6 ; //filler bits
  unsigned char
                     m_fillerbyte
                                        ; //filler bytes
};
```

Pretty obvious that this contains information about the players at the table. You can access balances / currentbets /cards (if yours or at showdown) / names.

prw1326 Datastructures

You can find all information and examples on it in these articles: http://www.maxinmontreal.com/wiki/index.php5?title=Prw1326, http://www.maxinmontreal.com/wiki/index.php5?title=Enhanced Prwin.!!!

Examples

Acessing information from the "struct" "holdem_state"

```
double process_query(const char* pquery){
  if(pquery==NULL)
    return 0;    //example
  if(strcmp(pquery, "dll$test")==0)
    return 101;
  if(strcmp(pquery, "dll$dealerchair")==0)
```

```
return m_holdem_state[(m_ndx)&0xff].m_dealer_chair;
if(strcmp(pquery, "dll$currentbet3")==0)
   return m_holdem_state[(m_ndx)&0xff].m_player[3].m_currentbet;
return 0;
}
```

You can see in the code example above, how you have to use the point operator on a struct to access the information stored in the struct of the scraped state.

A function returning the flush rank of the player

For convinience we restructure the code, by adding another header file:

Click on project -> Add New Item -> select "Header File (.h)" -> name it
"OH-general", you can also do it in the solution explorer by clicking on the
"Header Files" folder.

```
//card macros
#define RANK(c)
          ((c>>4)&0x0f)
#define SUIT(c)
               ((c>>0)&0x0f)
#define ISCARDBACK(c)
               (c==0xff)
#define ISUNKNOWN(c)
              (c==0)
//consecutive states
holdem_state m_holdem_state[256];
unsigned char
          m_ndx;
//versus list & prwin
phl1k_t m_phl1k;
pp13 prw1326;
//WH symbols
pfgws_t m_pget_winholdem_symbol;
double gws(int chair, const char* name, bool& iserr) {
 return (*m_pget_winholdem_symbol)(chair, name, iserr);
}
double gws(const char* name) {
 bool iserr;
 int mychair = (int)gws(0, "userchair", iserr);
 return gws(mychair, name, iserr);
}
```

```
bool hlset( int rank0, int rank1, int listnum, bool onoff ) {
  return ((*m_phl1k)[listnum][rank0-2][rank1-2] = onoff);
}
bool hlget( int rank0, int rank1, int listnum ) {
  return ((*m_phl1k)[listnum][rank0-2][rank1-2]);
}
```

- Cut / paste the code above from the "OH-DLL.cpp" file to the "OH-general.h" header file.
- We will also need another include now, its a standart library of c++, wich contains a datatype called bitset.
- The bitset datatype converts integer values to their binary representation, we will need this because we are going to work with "srankbits".
- You can read more about how rankbit values are obtained in the chapter about OpenHoldem symbols; look at the very bottom of the page.
- You can find documentation on the C++ library here: http://www.cppreference.com/wiki/stl/bitset/start.

After you're done with cut-pasting add the "bitset" include and also "using namespace std;" this is neccessary line, don't think much about it, but the include would not work without it. Your OH-DLL.cpp "#includes" section should look like this now:

```
#define WHUSER_EXPORTS
#include "OH-DLL.h"
#include <windows.h>
#include <bitset>
using namespace std;
#include "OH-general.h"
```

Now we can begin with the actual coding, in the next step open "OH-general.h" and add the following code:

```
int srankhiplayer;
int srankbits;
int flushrank;
int set_flush_rank(){
bitset<16> srb( srankbits );
srb = (~srb ) >>= srankhiplayer;
return (int)srb.count(); }
```

This function will determine the flush rank via some bitset operations, 1 =

```
nutflush, 2 = 2nd nutflush, etc.
void int oh symbols(){
  srankbits = (int)gws("srankbits");
  srankhiplayer = (int)gws("srankhiplayer");
  flushrank = set_flush_rank();
}
This function will update the values each time a new state is scraped.
You also need to modify the "OH-DLL.cpp" to include this new symbol:
double process_query(const char* pguery) {
  if(pquery==NULL)
    return 0;
                //example
  if(strcmp(pguery, "dll$test")==0)
    return 101;
  if(strcmp(pquery, "dll$dealerchair")==0)
    return m_holdem_state[(m_ndx)&0xff].m_dealer_chair;
  if(strcmp(pquery, "dll$currentbet3")==0)
    return m_holdem_state[(m_ndx)&0xff].m_player[3].m_currentbet;
  if(strcmp(pquery, "dll$flushrank")==0)
    return flushrank;
  return 0;
}
double process_state(holdem_state* pstate){
        if(pstate!=NULL){m_holdem_state[ (++m_ndx)&0xff ] = *pstate;
        int_oh_symbols();
        return 0;
}
```

If all went well you can access the flush rank with "dll\$flushrank" on formula level now.

Files

The files for this tutorial can be found here: http://www.maxinmontreal.com/wiki/index.php5?title=Image:OH-DLL-B.rar.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-28T23:15:59.109000

Setting up the *.ohf File for DLL Use

This part of the series will concentrate on how we will pass decisions from the dll to the autoplayer. It's simple, the *.ohf file contains the instructions for the autoplayer and will call the variable from the dll which contains the decision.

Setting up the DLL

First we need to introduce a decision, variable. You can download the files used for this part. I will go through the changes I made:

- Added includes for preflop, flop, turn, river code
- Added include for global vars
- Added decision variable and call in "process_query"
- Added code to set the decision variable in "process_state"

Changes to OH-DLL.cpp

Include Section

The include section looks like this now:

```
#define WHUSER_EXPORTS
#include "OH-DLL.h"
#include <windows.h>
#include <bitset>

using namespace std;

#include "OH-globalvars.h"
#include "OH-general.h"
#include "preflop.h"
#include "flop.h"
#include "turn.h"
#include "river.h"
```

Changes to process_querry / process_state

```
//example
  if(strcmp(pquery, "dll$decision")==0)
    return decision;
  return 0;
}
double process_state(holdem_state* pstate) {
  if(pstate!=NULL)
                         m_holdem_state[ (++m_ndx)&0xff ] = *pstate;
  //resetting decision
  decision = 0;
  //updating vars
  int_oh_symbols();
  //calculating decision if it's my turn
  if(ismyturn){
    switch(betround){
    case 1:
     decision = preflop();
     break;
    case 2:
      decision = flop();
      break;
    case 3:
      decision = turn();
      break:
    case 4:
      decision = river();
      break;
    default:
      break;
    }
  return 0;
```

You can already see from the comments what is happening here, each time a new state is scraped:

- "int_oh_symbols" updates the values of our defined oh-symbol variables.
- the "decision" variable is reset to "0".
- when it is our turn (buttons visible) we call the code depending on the current betround to set the "decision" variable.

Preflop, flop, turn, river includes

Until now they do not contain any code, they simply contain a function each which returns 0, e.g:

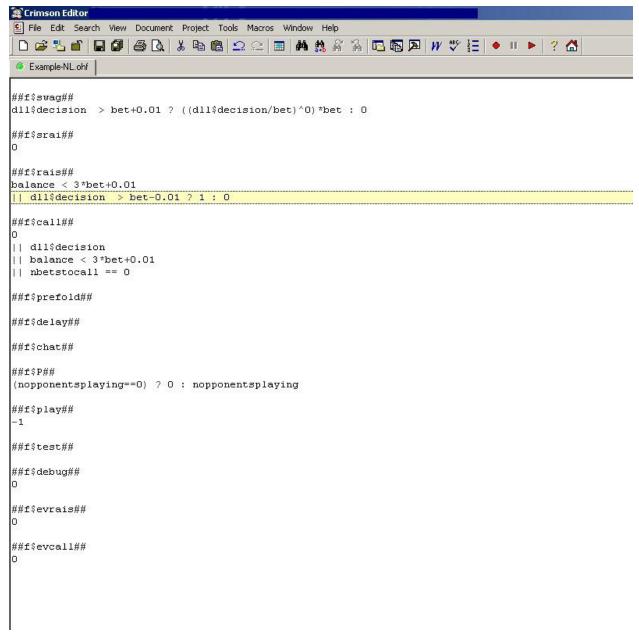
My next part of this series will be concentrating on coding a basic preflop for deep stack NL.

Include for global vars

```
//OH DLL VARS
//CONSECUTIVE STATES
holdem_state
        m_holdem_state[256];
unsigned char
        m_ndx;
//VERSUS LIST AND PRWIN
phl1k_t m_phl1k;
pp13 prw1326;
//SYMBOLS VAR
pfgws_t m_pget_winholdem_symbol;
//OWN VARS
//GENERAL VARS
int ismyturn;
int betround;
double decision;
//SUIT VARS
int srankhiplayer;
int srankbits;
int flushrank;
```

I decided it's best to keep the vars in one include, as you will have tons of code later and it will be much more convenient to organise it in include files. My former NL micro stakes script contained about 50k lines. As you can see I added two new variables, "ismyturn" and "betround".

Setting up the *.ohf file



This is how a *.ohf file might look for this dll, ofourse I am always open to suggestions if someone else has any to improve it.

Code in ##f\$betsize##

• If our "decision" is 1 bblind or more we will enter the closest amount in bblinds.

Code in ##f\$call##

- If for some reason we can't enter the betsize and your "decision" is not 0 we call.
- If the balance is smaller than 3 bblind we call. This is a failsafe, in case for some reason we will have put almost all money allin except a very small amout which can be left, if you are not very careful. Possible reasons are a problem with the exact dezimal representation of numbers or improper handling of the slider-bar.
- If we don't need to call anything we obviously check.

Files

The files for this tutorial can be found here: http://www.maxinmontreal.com/wiki/index.php5?title=Image:OH-DLL-C.rar.!!!

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-07-27T11:30:12.578000

Using prw1326

Included in the attachment is a "library" for prwin1326, it includes functions to set hands, manage weights, log prw1326 lists. Now i will not go into details to much simply take a look at the example below:

This is the main part of the code for this example:

```
double process_state(holdem_state* pstate)
      if(pstate!=NULL)
             m_holdem_state[ (++m_ndx)&0xff ] = *pstate;
//ACTIVATING PRW
prw1326->preflop = 1326;
      prw1326->useme = 1326;
//PRW EXAMPLE
int betround = (int)gws("betround");
      int userchair = (int)gws("userchair");
      int handnumber = (int)gws("handnumber");
      int ismyturn = (int)gws("ismyturn");
      if(ismyturn) {
             //list managment preflop
             if(betround == 1 && round_trigger != betround) {
                    //chair 0 -> VIP 50% PFR 10%
                    prw_set_preflop_list(0, 0.5, 0.1, 1024);
                    //chair 1 -> VIP 20% PFR 10%
                    prw_set_preflop_list(1, 0.2, 0.1, 1024);
                    //log preflop handranges
                    prw_log_handlist(handnumber, userchair, betr
                    round_trigger = betround;
             //list managment flop
             if(betround == 2 && round_trigger != betround)
             {
                    //chair 0 -> remove all pocket pairs
                    for(int i=0; i<13; i++)
                    prw_remove_pp(0, i, 1024);
                    //chair 1 -> remove all suited aces
                    for(int i=0; i<13; i++)
                    prw_remove_s(1, 12, i, 1024);
                    //log flop handranges
                    prw_log_handlist(handnumber, userchair, betr
                    round_trigger = betround;
```

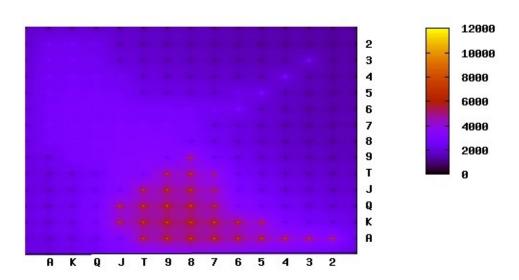
```
}
return 0;
}
```

All this piece of code does is:

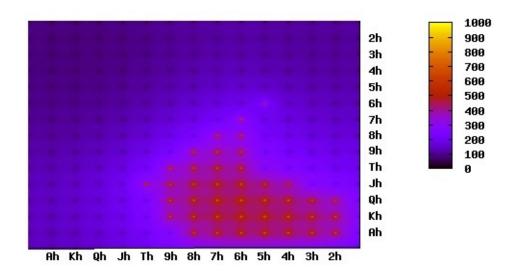
- 1. set up hand range for chair 0 which include 50% of the best hands and exclude 10% of the best hands, same for chair 1 with 20% and 10%
- 2. on the flop it removes pocket pairs out of chair 0's range and suited aces out of chair 1's range
- 3. creates a log file preflop
- 4. creates a log file on the flop

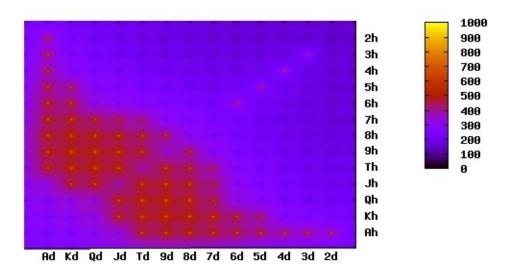
The log files are saved in "C:/prwlog/handnumber/" folder, this is how the log files look like, once they have been converted with gnuplot into images:

Handrange (chair: 0, betround: 1, log: 0)



This is a log of all 1326 hands, on the top left are the suited hands on the bottom right are the unsuited hands.





These 2 are 2 out of 10 more logs seperated by all suit combinations. Now to obtain these log pics you need gnuplot, as the DLL saves a gnuplot script in the log folder which needs to be executed to create those images. it is pretty simple:

- 1. download gnuplot: http://www.gnuplot.info/.
- 2. install and run gnuplot
- 3. open the log folder "C:/prwlog/handnumber/gnuscript/" there, if you ran this DLL (vs 2 opponents ideally in chair 0 & 1), you should find 2 files which have the name "gnuscript-0" (preflop log) and "gnuscript-1" (flop log) you can simple drag them to the gnuplot window or simply select "file -> open" and browse to their location.
- 4. if you execute them it may take a while till the "jpgs" are created in the directory because these are height projections of 3d plots and depending on your CPU this can take some minutes.

you can also set the resolution lower if it takes to long. look for this line in "prw log.h":

```
gnuplot_obj << "set dgrid3d 500,500,1" << endl;</pre>
```

Replace the 500,500 values with lower one.

File

The files for this tutorial can be found here:

- http://www.maxinmontreal.com/forums/download/file.php?id=2338
- http://www.maxinmontreal.com/forums/download/file.php?id=2337

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-01-06T22:51:05

Extensions

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-26T16:51:32.625000

↓Mouse ↓DLL

The mouse DLL extension provides for a user-specified handling of mouse actions. The standard, reference mouse.dll that is shipped with OpenHoldem performs very un-human-like mouse activities. For example, instantly warping the mouse pointer to a location on the button and clicking is unlikely to be human behavior. A human is more likely to start the mouse moving, accelerate to the target, then decelerate as the pointer nears the position of the button. Simulating mouse movements that are more human-like may help to avoid detection by casions.

The mouse DLL extension is available if you would like to implement custom mouse action behavior. The source code for the reference mouse.dll implementation can be found on Google Code in the source code repository. Two methods must be exposed in your mouse.dll in order to interface properly with OpenHoldem.

"MouseClick" is called when OpenHoldem wants to click a mouse button on a specific location on the screen. The prototype for the "MouseClick" method is:

MOUSEDLL_API int MouseClick(const HWND hwnd, const RECT rect, const

"MouseClickDrag" is called when OpenHoldem wants to click a mouse button at a specific location on the screen, hold it, drag the cursor to another location, and then release it. Dragging is from rect.left to rect.right, halfway between rect.top and rect.bottom. The prototype for the "MouseClickDrag" method is:

MOUSEDLL_API int MouseClickDrag(const HWND hwnd, const RECT rect, co

Parameter	Description
hwnd (in)	the HWND of the window to take the mouse action on
rect (in)	the bounding rectangle of the area to take action on — this generally corresponds to a Table Map Region record rectangle, like those that define button click areas; this rectangle is in relative client context the mouse button to be clicked, this is an enum:

enum MouseButton

```
MouseLeft,
button (in)

MouseMiddle,

MouseRight

A clicks (in)

restore_focus (in)

the number of clicks of the specified button
the HWND of the window to return focus, active and
foreground to, after the action has been taken; if this is NULL,
no focus resotre is requested
```

a POINT to return the cursor to, after the action has been taken;

if POINT.x and POINT.y are set to -1, then no cursor return is

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-29T22:17:53.859000

requested

restore_cursor

(in)

↓Keyboard ↓DLL

The keyboard DLL extension provides for a user-specified handling of keyboard actions. The standard, reference keybaord.dll that is shipped with OpenHoldem performs very un-human-like keyboard actions. For example, there is no delay between keystrokes. When entering a betsize, a human is likely to have somewhat random delays between keypresses as the betsize amount is entered into the betsize box. Simulating keyboard actions that are more human-like may help to avoid detection by casions.

The keyboard DLL extension is available if you would like to implement custom keyboard action behavior. The source code for the reference keyboard.dll implementation can be found on Google Code in the source code repository. Two methods must be exposed in your keyboard.dll in order to interface properly with OpenHoldem.

"SendString" is called when OpenHoldem wants to send a string to the poker client. Generally, this method is expected to click in the specified rectangle first, to activate it. The prototype for the "SendString" method is:

KEYBOARDDLL_API int SendString(const HWND hwnd, const RECT rect, con

"SendKey" is called when OpenHoldem wants to send a single key to the poker client. Generally, this method is expected to click in the specified rectangle first, to activate it. The prototype for the "SendKey" method is:

KEYBOARDDLL_API int SendKey(const HWND hwnd, const RECT rect, UINT ν

Parameter	Description
hwnd (in)	the HWND of the window to take the mouse action on
rect (in)	the bounding rectangle of the area to click in to activate – this generally corresponds to a Table Map Region record rectangle, like the chat box or the betsize field; this rectangle is in relative client context; if the members of the RECT struct are {-1,-1,-1,-1} then do not click in the rectangle first
s (in)	the string to be sent to the poker client
use_comma	if true, then use a "comma" instead of a "dot" as the decimal
(in)	separator
vkey (in)	the VKEY code of the key to send to the poker client

restore_focus (in)	the HWND of the window to return focus, active and foreground to, after the action has been taken; if this is NULL, no focus resotre is requested
restore_cursor (in)	a POINT to return the cursor to, after the action has been taken; if POINT.x and POINT.y are set to -1, then no cursor return is requested

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-07-27T11:34:42.687000

↓PokerTracker Query Definitions ↓DLL

As you might have noticed a new DLL is part of OpenHoldem since version 4.2.0: PokerTracker_Query_Definitions.DLL.

http://code.google.com/p/openholdembot/source/browse/trunk/PokerTracker_Que This DLL provides the ability to extend the set of PokerTracker-stats easily. The interesting part of the DLL is the file PokerTracker_Queries_Version_4.which h contains the definition of a struct:

```
typedef struct t_QueryDefinition {
    CString name;
    CString description_for_editor;
    CString first_part_of_query;
    // The queries will be created on the fly,
    // for both ring-games and tournaments.
    // There will be an "infix" added, depending on game-type
    CString last_part_of_query;
    int stat_group;
    bool needs_infix_and_second_part;
};
```

- "name" should be self-explanatory. It is something like "icon" or "vpip".
- "description_for_editor" is the explanation that will be shown in OpenHoldems formula-editor.
- "query" is is the database-query, usually containing one or more 3 place-holders %GAMETYPE%, %SITEID% and %SCREENNAME%.
 OpenHoldem will build the complete query on the fly and automatically replace these place-holders by the correct values. A nice side-effect of this new concept: all stats will automatically work for both ring-games and tournaments; therefore OpenHoldem does no longer distinguish between "pt_" and "ptt_" stats, simply "pt_" will always work.
- "stat_group": this is one of pt_group_basic / pt_group_advanced / pt_group_positional. Diffferent groups of stats need different sample-sizes to become meaningful or change significantly. Therefore OpenHoldem will evaluate these symbols with different update delays depending on the number of hands a user has played to achieve a good compromise between meaningful stats and a low database-usage.

And all the rest happens automagically. To add a new stat you only have to change the constant

```
const int k_number_of_pokertracker_stats = 29;
and append your stat to the array
t_QueryDefinition query_definitions[k_number_of_pokertracker_stats]
    /* PT4 query to get icon */
    {
        // name
        "icon",
        // description_for_editor
        "Poker Tracker auto-rate icon code",
        // query
        "SELECT val_icon as result \
               player \
        FROM
        WHERE id_site = %SITEID% AND \
        player_name LIKE '%SCREENNAME%'",
        // stat_group
        pt_group_basic
    },
/* PT4 query to get number of hands in the database */
        // name
        "hands",
```

After the coding work has been done you need to compile the DLL and put it together with the linker-file (*.lib) into the OpenHoldem directory and everything will work automatically: querying the database, accessing the values in your bot-logic, correct update delays, even the symbol-descriptions in the formula-editor. Despite of its simplicity this approach has some more benefits:

- it is update-safe. If a new version of OpenHoldem gets released your DLL will work as before without you having to compile a modified version of OpenHoldem or missing important bug-fixes.
- you can add stats without having to wait for new releases.
- you can easily contribute to make OpenHoldem better for everyone.

If you need more info about PokerTracker stats or some help you will find it in our PokerTracker sub-forum:

http://www.maxinmontreal.com/forums/viewtopic.php?f=273.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-26T16:58:59.140000

↓Chat

OpenHoldem's chat feature is provided for two reasons: deception and fun. The key to chat is to use it wisely and to not overdo it.

OH-Script

Create a new formula f\$chat with OpenHoldem, which returns 0, if you wish no chat and one of the constants described below to send a simple chat message. These constants are not available as OpenHoldem symbols, you have to use the numerical values (gg=1, nh=2, ...).

Available chat messages

A complete list of available symbols for formula

```
enum {
    no_simple_Chat = 0,
    simple_Chat_gg,
    simple_Chat_nh,
    simple_Chat_vnh,
    simple_Chat_n1,
    simple_Chat_wow,
    simple_Chat_lol,
    simple_Chat_rofl,
    simple_Chat_haha,
    simple_Chat_tu,
    simple_Chat_thx,
    simple_Chat_omg,
    simple_Chat_sick,
    simple_Chat_fu,
    simple_Chat_fish,
    simple_Chat_hi,
    simple_Chat_hello,
    simple_Chat_bye,
    // 18 :)
    simple_Chat_happy,
    // 19 :(
    simple_Chat_sad
    // 20: \o/
    // 21: /0\
};
```

OH-Script with DLL

Simply define f\$chat as dll\$chat and handle this symbol. Return the same constants as used above to specify the message to send, especially "no_simple_Chat" to send no message.

DLL only

Using Chat in your DLL gives you two additional opportunities: Any chat message you want At any time

At startup your DLL will receive an additional message "p_send_chat_message", providing a pointer to access the chat function. The function has the following prototype:

```
void SendChatMessage(char* new_message);
```

Safety measure: To prevent a crazy bot flooding the chat, there are two settings in {Preferences}:

- Minimum delay (seconds)
- Additional random delay (seconds)

Adjust these to any value you are comfortable with, if you trust your bot.

Chat and Table Maps

In order to use the chat function, OpenHoldem has to know where the chat box of your casino software is. Therefore the corresponding TableMap has to specify a region that defines the location of the chat box. transform.

Without the chat box region, all attempts to chat will be ignored without any warnings, avoiding anything that could disturb the screen scraping process.

Limitations

- One message at a time. There is no waiting list, or queue, to handle multiple messages (as we are lazy and assume, there's no need for too much chat at a poker table).
- There is no way to specify any concrete delay. The messages will be sent only when {The Autoplayer} is active and there will be an additional delay per character, the intent being to simulate human-like input.
- Only predefined chat messages if you are not using a DLL.
- No support for non-alpha-numeric-characters.
- On some sites that temporarily block the chat (e.g. Cakepoker), multiple chat messages can queue up in the poker client. The obvious counter measure would be to always delete the chat box, but that could raise a red flag on other sites.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-09T15:06:40.125000

Advanced Topics

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:17:12.296000

Automation

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-03-22T21:18:56.703000

↓ Automation

Alright, so you have made it through the previous sections, and have a working Table Map, a set of bot logic for your target game, and a fully-stealthed out configuration. You may have just started to realize what a pain it is to have to manually start the software, start the casino, choose a table, sit at the table, leave the table, start a new table, etc. This becomes even more oppressive if you want to run multiple simultaneous tables.

The answer is automation. Automation will do these opening, selecting, moving, and clicking activities for you, and allow you to ultimately sleep while your poker bot plays for you. Nothing better than making money while you sleep!



Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-09T15:45:09.648000

Planning a Hopper

Creating a bot that wins (or buying a bot that makes the botter whine) is one thing. Putting it to the tables regularly to make some volume is another task that can be time-consuming and boring. Of course this one can be automated too, but only some tasks of the job are directly supported by OpenHoldem. OpenHoldem is mainly designed as a single-tabling poker-playing application. It is not designed for handling multiple windows (lobby, registration popups), scheduling sessions or interpreting non-numerical input. In this chapter we will consider the tasks that usually need to be done by a hopper.

Preparatory task

- Start the casino
- Take care about potential casino updates
- Log in with username and password
- Prepare / verify correct settings
- Prepare the lobby (game-type and level)

Main hopper loop

- Select new tables (they might or might not open immediatelly, e.g if you register to a tournament)
- Buyin with the desired amount (once the table is open)
- Choose an empty seat
- Supervise the number of open tables (including registered tournaments that are expected to open soon)
- Manage the session time (open tables until end of session x)

Final tasks

- Wait until all tables are finished and closed
- Close the lobby
- Close all instances of OpenHoldem (if desired)

Tasks that can be done by OpenHoldem

- Automatically start new instances of OpenHoldem for popup-blocking and for the next table (since OpenHoldem 8.0.1)
- Connect to new tables that do not yet get served by other instances of OpenHoldem
- Arrange the poker-tables in a non-obstructive way so that every table can be scraped properly
- Close popups (true windows with the popup-blocker, painted casino-advertisements that look similar to windows with the i86-spam-buttons)
- Tick "wait for big-blind" in cash-games
- Tick "auto-post blind"
- Click "sitout" / "leave", e.g based on session-time, opponents and PokerTracker stats.
- Click "rematch"
- Execute a custom rebuy-script if f\$rebuy returns a positive number (rebuy-amount in dollars)
- Click the [x] in the title-bar to close the poker-table if f\$close is true
- Disconnect if the table disappears and be ready for the next table
- Shutdown OpenHoldem, using the f\$terminate-function (since OpenHoldem 8.0.1)

Architectural conclusions

As you can see a hopper can be devided into four functional parts

- Starting and configuring software
- Closing software
- Starting new tables, the main hopper loop
- Managing a connected table (singular!)

The first two jobs are more or less trivial. Opening new tables in the main hopper loop is the main part of a hopper that can't be done by OpenHoldem (designed as: single bot, single table, single window). So all you truely need nowadays is a *table-opener* (fire and forget). There are cheap and good table-openers available for the main casinos that get used by flesh-and-blood poker-players. Only for smaller pokersites you have to create such a thing on your own. All the remaining tasks (posting blinds, rebuys, closing the table) can be done with OpenHoldems built-in hopper functionality that got added over time. This is not too difficult and only requires the configuration of a handful hopperfunctions like f\$sitout and f\$close, plus some casino-specific tablemap-regions

(sitout button) to make it work everywhere. Finally a small example to demonstrate the simplicity of OpenHoldems hopper-functions:

```
##f$close##
WHEN (issittingout
   AND handsplayed > 0
   AND elapsedauto > 120) // No autoplayer action since 2 minutes
RETURN TRUE FORCE
```

.

AutoIT

The general tool of choice for automation is AutoIt (http://autoitscript.com).



AutoIt is a free (as in beer — this is not an open source program) Windows scripting language and execution engine. From the web site (introduction part: http://www.autoitscript.com/autoit3/index.shtml):

"AutoIt v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI and general scripting. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g. VBScript and SendKeys). AutoIt is also very small, self-contained and will run on all versions of Windows out-of-the-box with no annoying "runtimes" required! AutoIt was initially designed for PC "roll out" situations to reliably automate and configure thousands of PCs. Over time it has become a powerful language that supports complex expressions, user functions, loops and everything else that veteran scripters would expect....AutoIt is a freeware Windows automation language. It can be used to script most simple Windows-based tasks... AutoIt has been in popular use since 1999 and continues to provide users and administrators with an easy way to script the Windows GUI. In February 2004 the latest version of AutoIt - known as AutoIt v3 - was released and added powerful scripting features."

Automation of poker botting is one of those "simple Windows-based tasks" that AutoIt is great for. The challenge is that now you have a new language to learn (does it ever end?), but the support for AutoIt on their forums is astounding, and you can also get help on the OpenHoldem forums.

- AutoIt Links: http://www.maxinmontreal.com/forums/viewtopic.php?f=175&t=10520
- How can I download AutoIt?

http://www.autoitscript.com/autoit3/downloads.shtml Select: AutoIt
Full Installation

- If you need more tools to create code, download: http://www.autoitscript.com/autoit3/scite/downloads.shtml
- Where is AutoIt documentation? You can go to your AutoIt-folder and run AutoIt3Help.exe or take the Online Documentation (http://www.autoitscript.com/autoit3/docs/)
- First Steps in AutoIt If you are new. You can see "Hello World" in AutoIt: http://www.autoitscript.com/autoit3/docs/tutorials/helloworld/he

↓Scout - A Hopper

Hopper is the name generally given to AutoIt scripts that automatically start and connect poker windows for you. One of the formerly best (and most complex) hoppers is Scout, that is open-source just like OpenHoldem. It is availabe at http://scout-poker-bot-hopper.googlecode.com/files/Scout2.68.zip. Scout will not work out of the box, guaranteed, and it implements a bit "too much" functionality for modern OpenHoldem 7.x, but it can serve as a possible starting point for your script for your casino. This is an important point — hoppers, like Table Maps, are very specific to a given casino. Think about the actions you take to join a table on different casinos — the actions to do that all differ ever so slightly!

Avoid \(\text{race-conditions} \)

Race-conditions happen if two or more processes interrupt each other and try to access the same resource at the same time. For example it might be, that OpenHoldem tries to enter a betsize at its poker table, but then your hopper grabs the mouse and interrupts the bot in the middle of an action. All kind of unexpected results can happen.

The good news: there is a solution to this problem. OpenHoldem will take care of race-conditions automatically; it will synchronize the autoplayer-actions of all instances, including actions taken by the rebuy-function and by the chatfunction. It uses a system-wide ↓mutex to ensure exclusive access to the keyboard and the mouse. Unfortunatelly OpenHoldem is not the only program, which uses these resources; usually the hopper competes for them, too. So any good hopper should wait with its actions, until OpenHoldem has finished its input; it has to implement the same protection-mechanism and wait for the same mutex. The default name of that mutex is "OHAntiColl", but it can be changed in the preferences to your liking.

An example AutoIT-hopper-code has been provided by USBFreak (http://www.maxinmontreal.com/wiki/index.php5?
title=System_Wide_Mutex_Handling_-Sample_Code):

System Wide Mutex Handling - Sample Code

Below are 3 functions you'll need if you plan to avoid OH and hopper "collisions" over mouse and keyboard inputs. As you may know, OH ensures it can act on the table in a consistent manner by seizing control of mouse / keyboard input (system wide mutex), performing its actions (clicking buttons, entering the betsize in text field etc.) and then releasing mutex so other applications can perform their tasks.

This works fine as long as you run one or few OH instances, and do not have a very active hopper to manage them. Once you are past that point and start running many tables with as many OH instances, plus moving tables around, browsing lobby etc., a situation where OH may interfere with hopper operations in the middle of an important task occurs, resulting in very unpredictable behaviour.

To avoid such a possibility, you may want to implement mutex handling in your hopper using below code.

The idea:

- 1. Create mutex lock at the start of your hopper. This seizes control of keyboard / mouse and no other application can send any input using them until mutex is released/unlocked.
- 2. Find a good time in your hopper loop when game state is stable (no lobby browsing, table opening / closing etc.) and release mutex for a period of time (in my case this is (number of OH instances opened * 1) + 2 seconds). This allows OH instances to grab mutex control for themselves, perform any pending actions, and release mutex control again.
- 3. Once the predetermined period of time expires, create mutex lock again, and proceed with hopper operations knowing nothing can interfere with mouse / keyboard.
- 4. At the end of your hopper, release/unlock mutex.

Main hopper loop

```
;START YOUR HOPPER, CREATE MUTEX LOCK
MutexLock()
$Global_Mutex_Handle = _CreateMutex("OHAntiColl")
;ENTERING HOPPER LOOP
While (1)
    ;... HOPPER OPERATIONS
    ;... GAME STATE STABLE, UNLOCK MUTEX
    MutexUnlock($Global_Mutex_Handle)
    ; ... WAIT FOR OH TO PERFORM ACTIONS
    ;... TIME EXPIRED, CREATE MUTEX LOCK AGAIN
    MutexLock()
wend
; END OF HOPPER, RELEASE MUTEX
MutexUnlock($Global_Mutex_Handle)
; END OF HOPPER, CLOSE MUTEX HANDLE
Close Mutex Handle($Global Mutex Handle)
```

Hopper functions

```
Func _CreateMutex($sOccurenceName)
   Local $ERROR_ALREADY_EXISTS = 183
   $Global_Mutex_Handle = DllCall("kernel32.dll",
        "int", "CreateMutex", "int", 0,
        "long", 1, "str", $sOccurenceName)
   $lastError = DllCall("kernel32.dll", "int", "GetLastError")
   If $lastError[0] = $ERROR_ALREADY_EXISTS Then
```

```
Return SetError($lastError[0], $lastError[0],
            $Global_Mutex_Handle[0])
    EndIf
    Return SetError(0, 0, $Global_Mutex_Handle[0])
EndFunc ;==>_GetMutex
Func MutexUnlock($Global_Mutex_Handle)
    Local $releasemutex = DllCall("kernel32.dll", "int",
        "ReleaseMutex", "long", $Global_Mutex_Handle)
    Return (IsArray(\$releasemutex) And \$releasemutex[0] > 0)
EndFunc
         ;==> ReleaseMutex
Func Close_Mutex_Handle($hMutex)
    Local $aCH = DllCall("Kernel32.dll", "int",
        "CloseHandle", "hwnd", $Global_Mutex_Handle)
    Return (IsArray(\$aCH) And \$aCH[0] > 0)
EndFunc
Func _API_WaitForSingleObject($Global_Mutex_Handle, $dwMilliseconds)
    Local $aResult = DllCall("kernel32.dll", "long",
        "WaitForSingleObject", "long", $Global_Mutex_Handle,
        "long", $dwMilliseconds)
    Return $aResult[0]
EndFunc ;==> API WaitForSingleObject
Func MutexLock()
    While 1
        $Global_Mutex_Handle = _CreateMutex("OHAntiColl");
        If @error = 0 Then
            ;SUCCESSFULLY CREATED AND LOCKED MUTEX
            ExitLoop
        Else
            ;WE WILL EXAMINE MUTEX TO SEE IF IT IS RELEASED
            Local $result = _API_WaitForSingleObject(
                $Global_Mutex_Handle, 30000);
            ;WAIT FOR MUTEX FOR 30 SECONDS
            If \$result = 0 Or \$result = 128 Then
                SUCCESSFULLY LOCKED MUTEX
                ExitLoop
            EndIf
        EndIf
        ;WAIT A BIT BEFORE TRYING AGAIN
        ;WE ARE NOT PROCEEDING UNTIL WE HAVE CONTROL OF MUTEX
        Sleep(100)
    WEnd
EndFunc ;==>MutexLock
```

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-07-27T11:28:34.109000

Setting process CPU affinity - Sample Code

In some cases it may be beneficial to force poker client process to work off one CPU in multiprocessor machines (or dual VCPU VMs). Good example is OnGame, which uses much more CPU on dual CPU machines than on single CPU machines. This is of course a problem with their software but they are unlikely to fix it.

The downside is that poker software operations (browsing lobby, updates, table management, etc.) are now forced to work with one CPU, so multitabling (4+) may be taxing CPU0 - make sure to observe the behaviour with and without forcing affinity, and draw your own conclusions.

Here is the code that will set the CPU affinity of PokerClient_Exe to CPU0 (first CPU): Only one process should exist, \$PokerClient_Exe is your lobby.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-26T22:09:37.671000

Setting Windows Classic Display Theme

Below is some AutoIt code (reproduced from the Wiki: http://www.maxinmontreal.com/wiki/index.php5?
title=Setting Windows Classic Display Theme - Sample Code) to set your display / dekstop theme to Windows Classic. Typically used at the beggining of the hopper script to force Windows Classic theme, because XP can revert to XP style after reboot, which happens from time to time.

```
Local $Theme_To_Use = 'Windows Classic'
Local $ScreenSaver_To_Use = '(None)'
Run('rundll32.exe shell32.dll,Control_RunDLL desk.cpl,,5')
WinWait('Display Properties')
ControlCommand('Display Properties', ,
        'ComboBox1', 'SelectString', $Theme_To_Use)
; SELECT SCREEN SAVER TAB
ControlCommand('Display Properties', "", _
    12320, "TabRight", "")
ControlCommand('Display Properties', "", _
    12320, "TabRight", "")
While (ControlGetText('Display Properties', "", 1303)) = ""
   Sleep(100)
ControlCommand('Display Properties', _
, 'ComboBox1', 'SelectString', $ScreenSaver_To_Use)
ControlClick('Display Properties', , 'Button7')
$OptWTMM = Opt('WinTitleMatchMode', 4)
WinWait('classname=CoverWindowClass')
WinWaitClose('classnam=CoverWindowClass')
Opt('WinTitleMatchMode', $OptWTMM)
; DONE
```

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-26T22:09:50.625000

Rebuys

OpenHoldem is able to control external <u>trebuy-scripts</u> with the bot, as this provides several advantages compared to hoppers:

- easy synchronisation with other bots due to built-in mutex-handling (avoiding mouse- and keyboard-conflicts).
- perfect timing, e.g. only when it is not your turn or between two hands.
- no need for the hopper to take care about your balance and simplified hopper-architecture in general.

The rebuy-script

The process to rebuy is still very specific for each casino (similar to TableMaps and lobby-handling), so there is no way to provide a built-in method that works for every casino. This casino-specific part of the job has to be done by an external rebuy-script, provided by the user. But this rebuy-script will usually become quite simple: some mouse-clicks and maybe keyboard-input are usually everything that needs to be done. For an example and possible starting-point see "RebuyDemo" in the source-tree at GoogleCode..

Sanity-checks

Many things can go wrong, when you try to rebuy, be it incorrect timing or a rebuy to the wrong value. To avoid such cases OpenHoldem does provide some security measures:

- always acquire the autoplayer-mutex before executing the rebuy-script
- optionally (as specified in the preferences) require a change in hand-number between two rebuys
- optionally rebuy only, when the user doesn't hold cards
- optionally test heuristically for occlusion and sane balances
- optionally wait for a user-defined time between two attempts to rebuy

Calling the rebuy-script

Once all necessary conditions are met, OpenHoldem will execute the rebuy-

script, that got specified in the preferences, and pass the necessary information as command-line-parameters:

Commandline-parameters

Casino (name as specified in the TableMap)

Windowhandle of the pokertable (HWND)

UserChair (0-9)

Balance (in dollars)

SmallBlind (in dollars)

BigBlind (in dollars)

BigBet (in dollars)

Target-amount (in dollars)



Figure None.1 Rebuy

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-26T22:10:33.468000

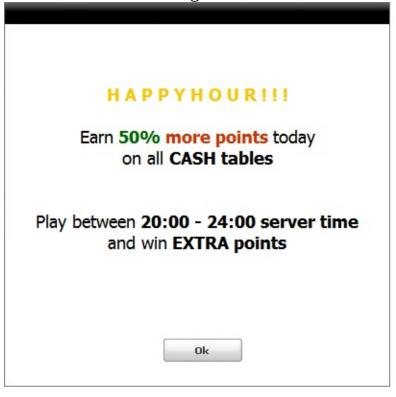
Handling Popup Windows

If you play at online-casinos for real money a couple of annoying things will happen now and then. Spam-messages will popup announcing tournaments, confirmation-popups will ask if you really want to leave or request other input. Third-party-software like anti-viruses might unexpectedly start running and open windows that overlap your tables

Of course you could create several external scripts, a complete hopper or even a multi-site hopper-framework to handle everything, you also could turn your anti-virus off like recommended in other forums (we don't recommend to do so). Thank god OpenHoldem already contains some tools that handle many hopper-tasks automatically or make it easy to deal with them with small effort, including the annoying popup windows.

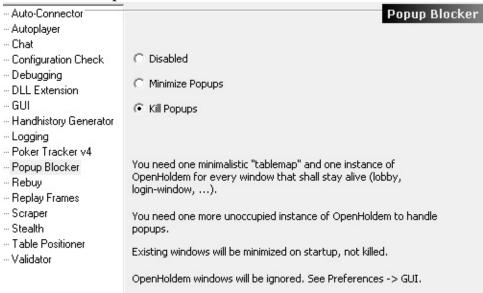
Messages that are painted on the table

Some casinos paint messages, mainly tournament-announcements, to the table. These messages are from a technical point of view no separate windows and therefore can't be handled by traditional popup-blockers. OpenHoldem tablemaps however support some anti-spam-buttons, that get clicked as soon as they get identified. These so-called i86X-buttons (i860..i869) require a clickable region and a button-state, but contrary to autoplayer-buttons no button-label, because the sole meaning of this buttons is "click it away".



The popup-blocker

Regular popups can be handled easily by OpenHoldems built-in popup-blocker. It will kill or minimize any visible window that is neither a connected pokertable nor an OpenHoldem window.



Keeping popups open

Sometimes you might want a popup to stay alive, e.g because you have to enter some data, like the amount of dollars you want to take to the table. In this case you can create a very minimalistic tablemap:

```
.osdb2
// OpenScrape 11.1.0
// 2017-01-28 18:47:19
// 32 bits per pixel
//
// sizes
//
z$clientsizemax 373 176
z$clientsizemin 373 176
z$targetsize 373 176
//
// strings
//
s$ispopup 1
s$titletext Buyin
```

- just clientsizemin/max and targetsize plus a titletext as usual for connection
- a symbol "ispopup" that disables the tablemap-completeness-checker and table-positioning. The popup will then stay at its ciurrent location and you can easily handle it, for example with a little script.

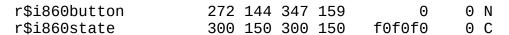
The same method is required to keep the lobby open if you use the built-in popup-blocker. In this case you should create a tablemap-symbol

```
s$islobby 1
```

Poker-lobbies receive special treatment. The table-positioner moves them to topleft of the screen (0, 0), where absolute screen-coordinates equal relative window-coordinates. This makes hoppers more robust in case a poker-table pops up and takes away the window-focus.

Klicking buttons

In some cases handling the popup might require very little work like clicking a confirmation-button. In this case the minimalistic popup-tablemap can be extended by an i86x-region.like the spam-handling example above. As soon as the button-state becomes true OpenHoldem will click the corresponding region.



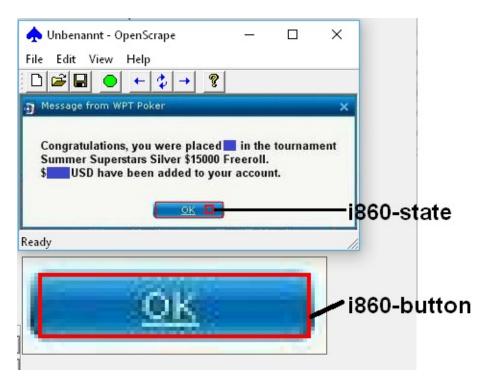


Table Changes in Tournaments

If you play multi-table-tournaments then you will be regularly moved to new tables when your opponents drop out and the table becomes too short-handed. From a technical point of view there are six cases how these table-changes can happen:

- The casino keeps the current window open, keeps the user in his current seat, only the opponents change. This case is completely non-critical and OpenHoldem will keep on playing.
- The casino keeps the current window open, but moves the user to a different seat. This case is not really critical either. As soon as OpenHoldem sees cards in a non-showdown-situation it will recognize the user-chair and then lock the new value for stability reasons.
- The casino closes the old table and opens a new one. No problem:
 OpenHoldem will disconnect and the auto-connector will connect to the
 new table as if it was a new session. Then OpenHoldem will continue
 playing like before.
- The casino doesn't close the old table, but opens a new one. No problem: the built-in auto-starter (since version 11.1.0) will take care that at least one free instance is always running and waiting for new tables.
- The casino opens a new table with a different table-theme, e.g. the final table at FullTilt Poker. Not much problem either. You probably need a second tablemap for the new table-theme. If the table-size and the table-title are identical then your tablemap needs some tablepoints so that the auto-connector can select the correct map. But that's it.
- The casino keeps the old-table, but changes the table-theme. Don't worry.
 OpenHoldem will check the tablepoints regularly and if they do no longer
 match for a certain time then OpenHoldem will disconnect from the table
 and connect again, this time with the correct tablemap. And then
 OpenHoldem will continue playing like before.

Long story short: changing table-themes need a little bit support by your tablemap, but otherwise OpenHoldem handles all table-changes the automagical way.

Stealth

Stealth

An important consideration for any botter is that of stealth. Many casinos will attempt to detect the OpenHoldem software on your computer and will either warn you or seize your account's funds if they find it. There are two general approaches to addressing this problem, a *Single System* solution and a *Dual System* solution. OpenHoldem provides a number of facilities for achieving both of these setups and obscuring the program from detection.

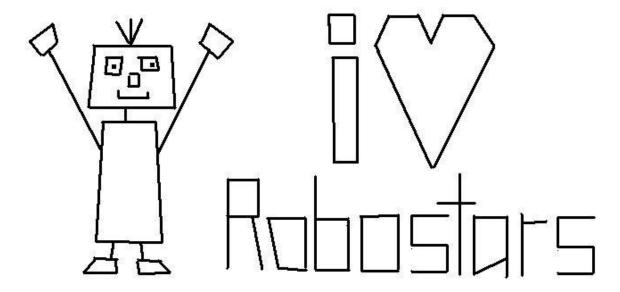
Note that a "System" as discussed in this chapter can refer to a physical machine or a virtual machine. VMware Workstation is a popular choice for consumer system virtualization, but certainly you can make use of the other consumer virtualization products available, such as VirutalPC or VirtualBox. Additionally, data center-class, bare metal hypervisor products such as VMWare ESX or Citrix XEN will work fine as well. See the VMWare section for information on configuring VMWare Workstation. Configuration of other virtualization platforms is not discussed in this documentation.

Your worst Enemy

In the next chapters we will talk about "stealth", hiding the fact that a bot plays at your account(s). For most newbies this means a sophisticated technical setup. But before we go into these technical details we should consider the real threats for your botting career.

The RoboStars Security Guard?

True, *some* casinos care about botters, some a bit more and some a bit less. But most casinos accept and even like botters, because they usually are good customers:



- botters provide traffic and playing opportunities for the recreational players
- they usually extract less money from the eco-system than a good winning player (these greedy evil 2+2 gangsters)
- some of the botters even bring money to the table. Consider all the stupid newbies wo buy a "winning profile" from a questionable salesman and then take it to the NL 100 tables. There have been thousands of them in the past at various botting sites and forums.
- the few long-term botters generate lots of rake for the casino, sometimes even more than their own winnings
- they are handsome and don't cause much work for the support team: no questions and no problems (well, as long as the opponents don't complain...)

Some casinos like botters so much that they actively seek cooperation. We know at least one small italian gambling site that got in contact with a commercial poker bot seller. The forum owner was enthusiastic — until it turned out that cashouts took at least six months — if they appeared at all. The cooperation

never happened.

The Players at your Table?

Getting reported by your opponents is a serious threat. Then casinos have to react, no matter how much they like botters. Some casinos tried to reduce this risk and disabled support for tracking-software; others even invented anonymous tables to protect their fish — and to protect their botters too.

Considering these points it becomes obvious who is your worst opponent:

You!

The main threat for your botting-success is your own bot-like behaviour. In a certain botting-forum people advise to play "not more than human-like 12-16 hours a day" and newbies ask if they can play "human-like 18 hours" too. Others run multiple cashgame bots with exactly the same stats. Simply sorting your opponents by PokerTracker-stats will reveal everything. It is always the same story:

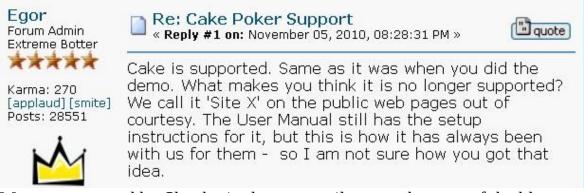
Most people fail for stupidity and greed.

Pokersites and Security

IPøker 💮 💮 💮	1)
Bovada / Bodog	1)
888	1)
Cereus	2)
OnGame	3)
Winning Network	4)
Prima Network	5)
MicroGaming	6)
Cake	7)
Merge	8)
PartyPoker	9)
FullTilt (now operated by PokerStars)	10)
PokerStars	11)

Remarks

- 1. IPoker, Bovada/Bodog, 888: supported by Shanky Bonusbots. Lots of people botting there naked without any stealth and insane "human-like" hours. But be careful. Even a bot-friendly site like 888 will ban you if you play 450 SNG each day like one of our users did recently.
- 2. Cereus (former Absolute and Ultimate): the scandal casino with the worst reputation on earth. Bots welcome.
- 3. OnGame: we once knew a guy who ran a farm there, 55 acounts, NL 10 SSS, singletabling (because his hopper was not able to run more than one table), playing all accounts naked round the clock. It took six weeks until he got busted.
- 4. Winning Network: no bans known
- 5. Prima Network: the only ban we ever heard of happened after a long holiday (bot turned off).
- 6. MicroGaming: occasional bans
- 7. CakePoker: quite bot-friendly. They disallow PokerTracker-support and have anonymized hand-histories to protect both fish and bots. When they got supported by Shanky BonusBots several years ago they asked Egor not to advertise that fact in public and get called "Site X" there, but hundreds of botters were able to play there naked until cake had to be unsupported for technical reasons. However they ban OpenHoldem botters (even with basic stealth), maybe because these bots are fewer and more sophisticated.



- 8. Merge: supported by Shanky in the past until a mass-ban out of the blue sky.
- 9. Party: supposed to be nearly as strict as PokerStars back in 2006 when Party got 70% of the Poker market. However reports about bans became very rare in the mean-time and more and more people report good successes.

- 10. FullTilt: always considered "yellow" in the early botting-years of WinHoldem. Stealth was recommended for users of OpenHoldem / WinHoldem, but botting was possible, whereas hundreds of Bonusbotters could play naked for several years, just like at Cake. Then out of the blue sky FullTilt banned all BonusBotters, Black Friday happened, the owners of FullTilt ran away with their customers money and PokerStars bought this casino. Strange enough: botting at FullTilt is still doable.
- 11. PokerStars: "red"; always strictly anti-bot. Botting there requires sophisticated stealth, though we know several people who bot there, each one with a highly individual self-made setup. If you are a newbie who runs a bot at PokerStars naked they will warn you once. If you try to hide a bot or get caught a second time they will ban you and seize all your funds. Botting there is recommended only for the brave and very experienced

Rules of Thumb

- Choose small sites. They usually don't have the man-power and know-how to go for a bot-hunt and they need every customer, including bots that provide traffic for the real players.
- Choose large networks. This advice might seem to contradict the first one, but large networks usually consist of several (and sometimes dozens) of small sites that share the player-pool. Each site is mostly interested in its own business; they don't have much man-power and know-how either; and the communication and cooperation between these sites is often nonexistent.
- Choose sites that are supported by "ready-to-go" poker-bots (i.e. Shanky BonusBots, as WinHoldem and OPI are dead and all other commercials we know about are plain scams). The reasoning is simple: if other people mostly with limited computer-skills can run a bot naked, then you should be safe too. Exceptions exist, see the list above.
- Stay away from large monolithic sites. Simple fact: PokerStars, FullTilt and PartyPoker have the man-power and the know-how to go for a bot-hunt. They don't need the traffic nor the rake, but they have much to lose in case of a **itstorm on twoplustwo.com. So these sites care, some a bit more, some a bit less. Stay away as a beginner.

The Grand Plan

So you want to start real money botting? Knowing your worst enemy you can plan a practical stealth setup and avoid all unnecessary complexity. In our opinion the way to success looks like that

- Create a basic bot, that plays something that looks like poker. ManualMode, DDPoker, PokerTH and PokerAcademy are great ways to develop your first basic bot offline, without having to care about stealth, hoppers and table maps at all (honestly: the TM for PA might need some tweaking, but that's it).
- Once you have created a basic bot, fix or create a tablemap and start
 playing real opponents at a site, that does not care. Your first version will
 have some bugs and maybe simplified strategy. It won't win you any money
 and you shouldn't use a real account for that bot (just in case). Register as
 "Mickey Mouse, Mouseton", take a paysafe-card and cash some money in
 and start playing and improving.
- Once you have a winning bot at a meaningful samplesize you can start creating a farm. You still should choose a site that doesn't care, but now you should care about stealth (at least a bit). Now it is time to learn about standard stealth and setup a clean system.
- Even when you have managed to create a profitable bot, you should still forget about sites like PokerStars, PartyPoker and FullTilt in the early stages of building a bot-farm. These sites do care (some a bit more, some a bit less) and they will bust beginners (you still are a beginner stealth-wise). But the other poker sites provide enough traffic in the microstakes and low-stakes to keep you busy for some time and to master the basics of stealth.
- Once you have reached that point you will be a forum-veteran and a honourable poker-botter. Then Max will invite you to a beer to Montreal and you will be welcome to ask any question you like.



FAQ

How do I cash out the money from the Mickey-Mouse-account? Forget about it!

If you want to make big money you have to invest big money. (Nick Coldhand)

This test-account will lose. Once your bot wins you still need an account to test future improvements and you don't want to risk your real accounts just because of 50 bucks. Always remember: *Most people fail for stupidity and greed*.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-05-10T21:15:00.406000

OpenHoldem JSingle System Configuration

In this scenario, both the poker client software and OpenHoldem reside on the same machine. In this case, reference the Stealth Preferences and Ninja Folders sections at a minimum. Review of the Rootkits section is recommended, also the section about Uninstallation if you need to clean up a real PC with was formerly used without any stealth.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-12T23:11:37.062000

↓Casino software

If possible try to install and run the casino with a normal, i.e. restricted user-account and not as administrator. If the casino software demands admin privileges to install/update/run, and if you've successfully configured your ninja-folder, then their software running as admin cannot automatically read your stealth drive(s) / folder(s) without physically taking ownership of the folder in order to gain access.

However most online casinos EULAs are notoriously vague when it comes to the measures they are allowed to take. The most openly discussed ones is looking in your browser history, or taking screenshots of your desktop. These are also both illegal in most countries but most casinos do it. For example in http://www.pokerstars.net/typeB_EN/poker/room/terms/ we see

5.6. You agree that PokerStars may take steps to detect and prevent the use of prohibited EPA Programs. These steps may include, but are not limited to, examination of software programs running concurrently with the PokerStars Software on the User's computer.

Until you know you are competent enough we suggest that you avoid networks with high security awareness. That said there are much easier ways for casinos to flag you and it is usually has to do with player greed (for example long hours, fixed schedule to the second, no breaks).

An advanced step (not covered in this article) is to install/enable file/folder monitoring/auditing that will allow you to track anything that happens to your "safe" folder. Note that doing this would not stop admin/casino software from taking ownership but you'd have a record of the event and would be warned which is better than nothing.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-19T20:37:15.796000

↓Ninja Setup

(Constructed from Tammelins posts here:

http://forum.winholdem.net/wbb/viewtopic.php?t=4749. and here http://forum.winholdem.net/wbb/viewtopic.php?t=11894_.)

This article explains the basic steps for creating a windows folder that will be as secure as it can possibly be under the NTFS file system.

The general goal is to create a folder that is owned by a single windows user account ("Ninja") such that no other user account may access the folder — not even the system or the admin account. Other users (including casinos that demand running as administrator) will also be unable to view the processes started by the Ninja-account. Therefore a Ninja-setup provides the same stealth like a rootkit, but it does not require any additional — and potentially compromising — software. The Ninja-setup achieves this level of stealth just with default Windows security-functionality.

Prerequisites

- 1. knowledge of creating/modifying user account
- 2. knowledge of windows explorer
- 3. knowledge of NTFS permissions

It is assumed that the reader understands how to create/modify a user account and how to use Windows Explorer to navigate folder space and how to modify folder properties/permissions.

User Accounts

First we create two user-accounts with limited permissions, one for the casino and the other one for the stealthed poker-bot ("Ninja").

- 1. Click "Start Control Panel User Accounts".
- 2. You had already set up one user while installing Windows. This user already has admin privileges. We'll call it the "admin" user. We need to give this user a password. Click on the user, then click on "create a password". Enter the password into the boxes and click on "create password". The next window asks you if you want to make your files and folders private. You certainly do, so click "Yes, make private". Click on "Back" to return to the user accounts window.
- 3. Now we'll set up two new users with fewer privileges. You will be logged in as the second user whenever you connect to a casino. We'll call this the "casino" user. The third user will be created to hide things from the casino. We'll call this the "Ninja" user.

Choose names and passwords for these users, and write them all down now
Admin user
Casino user
Stealth user

- 5. The casino user will have restricted privileges initially. Click "create a new account", enter the name of the casino user and click "next". Click on "limited" and then click on "create account". The casino user will appear next to the admin user. Click on the casino user and give it a password.
- 6. The stealth user will have restricted privileges permanently. Click "create a new account" and set up the stealth user in the same way you set up the casino user.
- 7. Click "create a new account". Choose a name and then click on "limited account".
- 8. That completes this step. If there is a "guest" user shown, you can ignore it. Close the "user accounts" window.

File System

- 1. Create the target folder (from admin account). In an ideal world you would create a top level folder on an empty local (non-shared/not-networked) hard drive where "Ninja" has full control over that disk. If this is not possible then the next best thing is a top level folder on a non-shared drive. If this is not possible then you must select a location where "Ninja" will have access to the parent folder so that it can access its private folder.
- 2. Permissions (from admin account)
 - 1. Select the properties for the folder and select the security tab add "Ninja" to the access list and grant "Ninja" full control. Remove all other users/groups from the access list, including the administrator(s), which will effectively shut out the entire world. There should be exactly one account in the access list and that account should be "<host>/Ninja"
 - 2. Uncheck/unselect "Allow inheritable permissions from parent to propagate to this object". The reason for this is that you want this folder and it's permissions to be the root node for all children below it. If you do not do this then permissions defined for the parent above your root folder will filter down to your folder and the children in your folder and you do not want that. Note that you cannot give ownership of objects in NTFS, you can only "take" ownership with the account you're currently using. You will take ownership of the folder later when you login to the "Ninja" account.
- 3. Logout of the admin account
- 4. Login to your "Ninja" account
- 5. Ownership (from Ninja account)
 - 1. Select the properties for your "safe" folder and verify that "Ninja" is the only account in the access list.
 - 2. Click the advanced button and goto the owner tab and select the "Ninja" account.
 - 3. Check "Replace owner ..." and click "Apply". You should now be the full owner of your "safe" folder and everything below it.
- 6. Permissions (from Ninja account)
 - 1. Click the permissions tab and check "Reset permissions on all child objects ...".
 - 2. Uncheck "Allow inheritable permissions from parent ..." (note this

should already be unchecked since you did that from the admin account), then click apply. This will go quickly if the folder is empty. It can take seconds or minutes or more depending on the children folder tree below you.

- 7. Verify. Create a test child folder inside your "safe" folder, view the properites and verify that "Ninja" is the owner with full control and that no other account has access. If this is not the case then you missed a step above (probably the "Allow inheritable permissions ...").
- 8. Logout of the "Ninja" account
- 9. Login to the admin account
- 10. Verify. Try to access the "safe" folder. You should not be able to access the contents of the folder nor modify the folder properties/permissions. As an admin you can still take ownership of the folder but until then the admin access is denied.

Other thoughts

If you have the resources and you want the best possible solution then you should dedicate an entire physical hard drive to your stealth department. That drive should not be the bootable system drive, but should be a secondary drive. You should consume all 4 primary partitions on your stealth drive by creating 4 primary partitions of roughly equal size. Each partition should be formatted with NTFS only (not FAT or FAT32).

You should then make the "Ninja" account the owner of all 4 partitions and the sole account with access. There should be no other accounts listed in the access list. The drives should not be networked or shared. If you do this you will notice that your admin account cannot even read the device label and that the 4 hard drives will be seen only as "local-drive". All access from any accounts other than "Ninja" will be denied.

The benefit of this setup is that once you construct a hard drive like this you don't have to worry about permissions ever again in the context of that drive you know that anything you do on that drive must be done from the "Ninja" account and that any folder/files created on that drive are entirely safe.

↓Stealth Preferences

By default, OpenHoldem uses names for window titles, mutexes, and class names that are easily detectable. It is highly recommended that you change these settings in Preferences to something non-obvious.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-12T23:14:15.687000

JIni-Files and Registry

By default OpenHoldem will store its preferences, settings and the list of last recently used files an Ini-Files to avoid any fingerprints in the registry.

```
OpenHoldem_Preferences__feel_free_to_rename_this_file_to_whatever_yo
```

The filename says everything: you can name this file however you want as long as there is only one such ini-file in the OpenHoldem directory. However: other applications such as OpenScrape still use the registry. So if you have installed other applications of the OpenHoldem product-suite on your casino-machine you might want to read the Chapter about uninstallation.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2014-12-26T21:23:14.531000

↓Rootkits

Rootkits are generally considered to be the subject of viruses, malware and other anti-social computer topics (http://en.wikipedia.org/wiki/Rootkit). In the case of poker botting, however, we can use rootkits to our advantage. Specifically, since rootkits can hide such things as viruses, they can also hide OpenHoldem programs, \(\perp\) registry entries, \(\perp\) network ports (\(\perp\) Bring traffic), files, etc. This document will not give you specific steps for configuring and using a rootkit. That is up to you. However, a Google of "hacker defender" may or may not turn up some interesting results.

Your anti-virus software will go nuts if you try to run a rootkit while your anti-virus software is operating. If you can configure your anti-virus software with a white list, then this may help the situation. Otherwise, you may need to uninstall your anti-virus software to use a rootkit.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-12T23:15:22.328000

OpenHoldem Dual System Configuration

In this scenario, the poker client software resides on "System A", and the OpenHoldem software resides on "System B". As the OpenHoldem software resides on a wholly different machine than the poker software, less thought needs to be given to protecting detection of the OpenHoldem software. OpenHoldem connects to the poker client windows using the Bring software as an intermediary. It should be obvious, but in order to make this scenario work, you either need two physical machines or two virtual machines. Rootkits still may be needed to obscure the network ports and traffic between the two systems.

Document generated by <u>eLyXer 1.2.5 (2013-03-10)</u> on 2015-04-12T23:13:02.593000

↓Virtualization

This is a guide to show you how to get OpenHoldem running securely on a single computer using the VMware Server application as a second, "virtual" computer. Original post by Tammelinn:

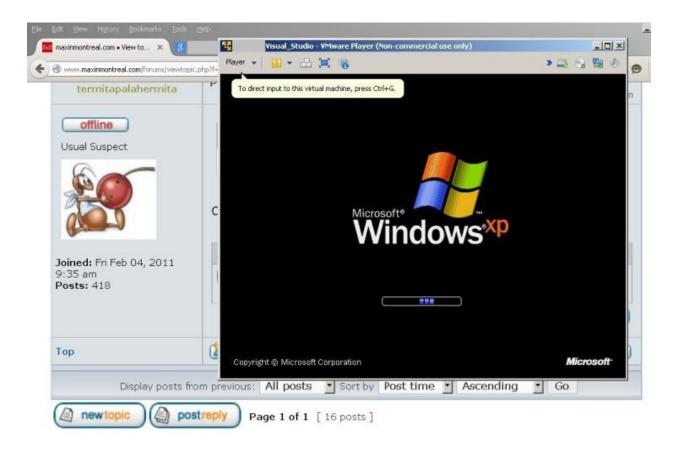
http://forum.winholdem.net/wbb/viewtopic.php?t=11894

I am also assuming that you are using the same operating system as me. If you are using Windows XP Media Centre Edition or Windows XP Professional, you can use this guide. If you are using some other brand of Windows (including Vista), the guide might not be reliable. I have only tested these instructions with XP. Most steps also apply to other virtualization software like (VirtualBox), which is a free and equally good alternative, but this guide concentrates on VmWare, which was state of the art when these instructions got written. The guide assumes no special knowledge, but you will need some basic computer skills. You will need to know how to install software, how to rename files, how to navigate folders and so on.

Much of this information is already available elsewhere. In many cases, all I have done is to reword it a little.

Definitions

The "real" computer is the one that is sitting on your desk or on your lap. It has its own operating system. The "virtual" computer is the one we are going to create using VMware Server. We can think of the "virtual" computer as a separate computer with its own operating system, even though it is physically part of the "real" computer.



Step 1 — Download VMware Server

VMware Server is a free download from http://www.vmware.com. You will need to provide VMware with a valid e-mail address to get a serial number. I used version 1.0.3 while writing this guide.

You will also need a Windows installation disk or an ISO-image of e.g. TinyXP. I am using a Windows XP Professional installation DVD, which I'm going to install as my "virtual" operating system.

While you wait for the download to finish, you should read the whole of this guide and make sure that you understand it. It would be a good idea to print it out, if you can.

Do not install any software until you are told to.

Step 2 — Install VMware Server on your real computer

Double-click on the VMware Server file you downloaded and it will begin installing itself. When you are asked if you want a "complete" or "custom" installation, choose "complete". A big window with some warnings about "IIS" might appear. We don't need IIS, so just click "OK". When the installer asks for

Internet access, grant it. Wait for the installation to complete before moving on to the next step.

Step 3 — **Configure VMware Player**

Double-click on the "VMware Server" icon that should have appeared on the desktop of your real computer. VMware Server will start for the first time. A Pop-up window asks you to choose between a local host and a remote host. Choose "local host" and click "OK".

Click the "New Virtual Machine" icon and a window will appear. Choose "typical" and click "next". Now choose which operating system you want to install on your virtual computer. Here, I selected the "Microsoft Windows" option, and the version I selected was "Windows XP, Professional". if you want to install from a CD-image and not from a real CD then just point VMware to the location of the ISO-image.

The next window gives you two boxes to fill in. The first box gives you a chance to name your new virtual computer. Choose any name you like.

The second box ("Location") allows you to choose where the Virtual computer will be saved on your hard drive. You probably don't need to change the contents of this box. (The virtual computer is actually a single file which exists on the hard drive of your real computer. If you want this file to be in a different folder or even on a different partition, you can tell VMware now.)

Next you must choose the type of ⊥network. Choose "Use bridged networking". The next window asks you to decide how big the virtual computer's virtual hard drive is going to be. The suggested 8GB is quite enough for the purposes of playing poker. Click "Finish" when you are ready. Some operating-systems like Windows 2000 or TinyXP will be satisfied with a bit less disk space, but don't forget that you also need space for casino-software, hand-histories and maybe PokerTracker. Better safe than sorry.

VMware will thereafter spend quite a few minutes creating the virtual computer. Wait until the installation is complete. You should be looking at a brand-new installation of Windows XP. If you can run Solitaire (click "Start - All Programs - Games - Solitaire") then you are ready to move on to the next step.

Step 4 — Install an ⊥operating system on the virtual computer

When VMware is ready, you'll see a new window with some details about the virtual computer. Find "Start this virtual machine" on the left and click it.

VMware will now look for the Windows installation disk in your CD/DVD drive. Put the installation disk in the drive and Windows will begin installing onto your virtual computer.

Soon, Windows will ask you if you want to format your hard drive. Only the virtual hard drive of the virtual computer is going to get formatted, so you don't need to worry about the data on your real hard drive. It is perfectly safe. If you asked how you want to format the virtual hard drive, make sure you choose one of the NTFS options. (Do not choose any option with FAT16 or FAT32). Eventually, Windows will ask you to enter your full name. We are going to create three seperate users, so choose a name for the first user ("Jessica", for example) and enter it here. This user will have permanent administrator privileges.

Next, enter an administrator password for this user. Write down the name and password now.

Administrator name and password _____

Later, Windows will ask you to enter up to five user names. The first user - the box labelled "your name" - should be the same as the user you entered above ("Jessica"). Don't fill in the other boxes yet.

Wait until the installation is complete. You should be looking at a brand-new installation of Windows XP. If you can run Solitaire (click "Start - All Programs - Games - Solitaire") then you are ready to move on to the next step.

Step 5 — Install VMTools, \downarrow Firewall and \downarrow Antivirus on virtual computer

VMTools is a package which, among other things, gets the mouse to move smoothly between the real and the virtual computer. You need to install it now, so press CTRL+ALT to restore mouse control to the real computer. Then, on the VMware Server window, click on the menu option VM, and then choose "Install VMTools..." VMTools will now install itself inside the virtual computer. The virtual computer will need to reboot.

After the reboot, find the VMTools icon in the system tray, next to the clock. It looks like three small interlinked squares. When you have found it, double-click on it.

A window appears and the "Options" tab is already showing. Find the line "Show VMware Tools in the taskbar" and click on it, so that there is no longer a tick next to it. Click "OK" to save your changes. This measure prevents the casino from seeing that you're using VMware - just in case they take a

screenshot of your desktop.

Now, your virtual computer is connected to the Internet, so it needs firewall software and antivirus software, just as your real computer does. Install the software (connect to the Internet to download it, if you have to), and then run a scan to check that your virtual computer is free from viruses.

(You could use AVG Antivirus (http://free.grisoft.com), and Zonealarm Firewall (http://www.zonealarm.com))

Step 6 — Check that the real and virtual computers can communicate

If the real computer and the virtual computer can't communicate with each other, all your efforts in securing them will be wasted. Let's check the communications now.

The traditional way to test a network is to have one computer "ping" the other. A ping is a short signal which one computer sends to another. The signal will get bounced back if the network is running as it should.

Before pinging anything, you need to turn off the firewalls on both computers. If you have installed a firewall such as Zonealarm, switch it off. Now click "Start - Control Panel - Security Centre". At the bottom of the window is an option to "Manage security settings for: Windows Firewall". Click on "Windows Firewall" and make sure it is off. Do this on both computers.

Now you need to find out the IP address for both computers. (An IP address is a numerical address which computers on a network use to identify each other.) On your real computer, click "Start - Run". Type "cmd" in the box and press return.

A new window appears. Type "ipconfig" and press return. You will see a series of numbers, some of which are marked "IP address". Find the one that corresponds to the type of Internet connection you have - for example, the one labeled "wireless connection" if you connect to the Internet through a wireless router.

The IP address of my real computer is 192.168.1.4 - yours probably looks almost
the same. Write it down now.
Real computer IP address
Now do the same thing on your virtual computer. Find the IP address and write it
down. The IP address of my virtual computer is 192.168.1.7 - yours will be
similar.
Virtual computer IP address

On your real computer, find the new window again and type "ping x.x.x.x", then press return. Replace x.x.x.x with the IP address of your virtual computer. You should see something like this...

```
Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.7: (and so on)
```

This dialogue means that the real computer sent a signal to the virtual computer four times, and each time it got a response.

If the network isn't set up properly, you will see something like this...

```
Pinging 192.168.1.7 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.7: (and so on)
```

This dialogue means that the real computer sent a signal to the virtual computer four times, but received no response.

Go to your virtual computer and do the same thing. This time, type "ping x.x.x.x" replacing x.x.x.x with the IP address of your real computer. If you get a response, your network is set up properly and you can continue to the next step. If the network isn't set up properly, you should seek help from someone who understands networks.

Don't forget to *turn your firewalls back on again* when you finish this step. If you have a good firewall such as Zonealarm, you can leave the less powerful Windows firewall turned off permanently (on your real computer as well as your virtual computer).

Step 7 — Ninja account

Now setup two restricted user-accounts. One for the casino-software. The other one for the bring-server that will handle the communication between your casino-VM and the bot-VM. The necessary steps are described in the chapter about single-system-configuration and Ninja-setups.

Step 8 — Install and configure Bring on your virtual computer.

Once you have created the two restricted accounts you should install the Bring server (but no other software) into the stealthed Ninja-folder. Bring will handle the communication with the bot-machine and it will receive and execute the actions on the poker-table. Installing and configuring Bring will require a couple of extra-steps that are not related to the VmWare-setup in general, so we made it an extra chapter (it should be the next one).

Use StringLet 1. Let 2. Let 2

"Bring" (http://www.winholdem.net/bring/index.html) is a software program developed by Ray Bornert. The sole purpose of Bring is to enable a dual-system configuration, and allow the casino's poker windows from "System A" to be displayed on "System B" where OpenHoldem can connect to them.

Step 1 — Install **J**Bring on your virtual computer

You now have a real computer and a virtual computer. OpenHoldem is going to run on the real computer. The casino software is going to run on the virtual computer.

Now you need to install some software that will allow OpenHoldem (on the real computer) to see what the casino software (on the virtual computer) is doing. There are several ways of doing this. We are going to use a piece of software called "Bring".

On the virtual computer, log in as the stealth user. Now go to the official WinHoldem site and download "bring.exe" (from the "stealth" page). Download it directly into the Ninja folder: http://www.winholdem.net/bring/.

The casino software can't see inside the Ninja folder, but it will be able to see Bring being run as a process. (If you want to see all the processes your computer is running, press CTRL+ALT+DEL and then click on the "processes" tab.) The solution is to rename Bring to something more innocent. Then, if the casino software looks at the processes you are running, it won't be able to find out anything incriminating about Bring.

Rename Bring now. For example, you could rename it to "notepad" or "paint".

Step 2 — Install <u>↓</u>software on your real computer

You can now install Bring on your real computer. You don't need to rename it; the casino cannot see what is happening on your real computer.

Open a folder on the desktop, call it "Bring", and download into it the files "bring.exe" and "bring.ini".

Click "Start - Control Panel - Folder Options". Click the "View" tab. In the box labeled "Advanced Settings", scroll down and find the line "Hide extensions for known file types". If there is a green tick next to this line, click on it to make the

tick disappear. Click "OK" to save your changes.

You should also install OpenHoldem on your real computer now if you haven't already. *Do not install OpenHoldem on your virtual computer*.

Step 3 — Configure Bring

Do you remember the IP address of your virtual computer? You're going to need it again now.

On your real computer, go into the "Bring" folder you just created and open the file "bring.ini" in a text editor. (Right-click on "bring.ini", choose "Open with..." and select "Notepad".)

Find the following lines:

```
# remotehost port password
a 192.168.x.x ???? prompt
b 192.168.x.x ???? prompt
```

Change them to this:

```
# remotehost port password
a x.x.x.x eeee secret
```

- x.x.x.x should be the IP address of your virtual computer.
- eeee is the "port" number. It can be a number of your own choice. 80 would be a good choice, and so would 800, 5000 or 8080.
- "secret" is a password. It's not a good password, but it is fine for the purposes of testing Bring.

Here are the changes I made to my bring.ini file:

```
# remotehost port password
a 192.168.1.7 80 secret
```

Step 4 — **Start Bring**

Now we're going to test Bring by playing a game of solitaire on the virtual computer, and by attempting to control it from the real computer.

Go to the virtual computer. You should still be logged in as the stealth user. Start the Solitaire programme ("Start - All Programs - Games - Solitaire").

Now open another command prompt (click "Start - Run", type "cmd" and press enter).

The command prompt window is currently looking at the "My Documents" folder. You need to move into the Ninja folder. If you followed the earlier suggestion, the location of your Ninja folder will be "C:\files". To move into this folder, simply type "CD C:\files" and press return. (If this doesn't work you will need to learn how to use the command prompt. Do an Internet search for "DOS command prompt".)

Now type the following command and press return. Instead of bring, you should type the name of your renamed file. Instead of 80, you should type the port number you chose in the previous step, and instead of "secret" you should type the password you chose.

bring -s 80 secret

A small pop-up will appear to show you that Bring has started on your virtual computer. Click "OK".

Now go to your real computer. Go into the Bring folder and double click on "bring.exe".

Another pop-up appears to show you that Bring has started on your real computer. Click "OK".

Another window appears labeled "Select remote connection". You should see a line that looks exactly like the line you entered in the "bring.ini" file.

a 192.168.1.7 80 secret

Click on this line so that it turns blue, and click the "OK" button.

Another window appears with a list of programs that are currently running on your virtual computer. One of them should be solitaire, so click on it so that it turns blue and then click the "OK" button. Solitaire will launch on your real computer, and if everything is set up correctly, you will be able to play it and watch your mouse clicks affecting both computers at once!

If you are unable to control solitaire from your real computer you definitely won't be able to control a poker table. Before you continue, seek help from someone who understands networks.

Step 5 — Stop running Bring

When you have had enough of playing solitaire, you should shut down Bring. On the real computer, just close the solitaire window. On the virtual computer, there are three ways to stop Bring.

- 1. Log off
- 2. Stop the Bring process. Press CTRL+ALT+DEL. Two task manager windows will appear, one for the real computer and one for the virtual computer. Find the task manager on the virtual computer and click the "Processes" tab, find "bring.exe" (however you renamed it), click on it to turn it blue, and then click on "End Process".
- 3. Lock the desktop. This doesn't actually close Bring, but it does halt it and prevent your real computer from communicating with it.

Stop Bring on both computers before you move on to the next step.

Step 6 — Set up the "run as a different user" facility

When you connect to the casino's Internet site you should always be logged into your virtual computer as the casino user. However, you will be running Bring *as if* you will logged on as the stealth user. In this way, the casino won't be able to detect that you are using Bring. Here's how to do it.

On the virtual computer, log in as the casino user. Right-click on the desktop and choose "New", then choose "Shortcut". A window will appear. In the box, type "cmd". Click "Next" and then click "Finish".

A new icon, "cmd.exe", will have appeared on your desktop. It's actually a shortcut to the command prompt that you have used a few times before. Right-click on this shortcut and choose "Properties". A window appears and the "Shortcut" tab should already be showing. Click the "Advanced" button. Find the line which reads "Run with different credentials". Click on it so that a tick appears next to it. Click the "OK" buttons twice. "Run with different credentials" actually means "run this program as a different user".

Start solitaire again. Double click the "cmd.exe" shortcut and a window will appear, asking you which user you want to use to run the program. Click on the line "The following user". Then, enter the stealth user's name and password, and click "OK".

The command prompt window appears. It is running as if you were logged in as the stealth user. We can safely run Bring from this window. Go into the Ninja folder (perhaps by typing "CD C:\files"). Type the following command, changing the words "bring", "80" and "secret" exactly as you did in step 4.

```
bring -s 80 secret
```

Bring's pop-up window appears, so click the "OK" button.

Now go to the real computer and start Bring there. Try to open the solitaire window, just as you did before.

Go back to the virtual computer. Press CTRL+ALT+DEL to make the task manager appear in both windows. Find the virtual computer's task manager and select the "Properties" tab. Check that "bring.exe" (however you renamed it) is running under the stealth user, even though you are logged in as the casino user. Check that "sol.exe" - the solitaire process - is running under the casino user.

Step 7 — **Check security with ↓Gaze**

We're going to make one more security check using the program Gaze. It can be downloaded from the forums at the WinHoldem official site.

(http://forum.winholdem.net/wbb/viewtopic.php?t=1114)

Still on the virtual computer and still logged in as the casino user, go to the WinHoldem site. There should be two files - "gaze.cpp" and "gaze.exe".

Download both of them to the virtual computer's desktop.

Double click on "gaze.exe" to run it. A window will appear, and shortly afterwards a pop-up labeled "Windows - no disk" will appear. Click the "Cancel" button, and close the Gaze window.

The results of the test have been saved to a file on the desktop called "gaze". Confusingly, there are now two text files on the desktop with the same name. Double click one of them to open it in a text editor. If the first few lines don't look like this, you've got the wrong file.

```
Sat Jun 23 17:40:42 2007
==
PROCESSES
==

PROCESS NAME: [System Process]
-- WARNING: OpenProcess failed with error 87 (The parameter is incor process ID = 0x000000000
thread count = 1 parent
process ID = 0x000000000
Priority Base = 0
```

You want to read the report on "bring.exe" (however you renamed it). Press CTRL+F to do a search for this file. When you find it, it should look like this...

```
PROCESS NAME: bring.exe
-- WARNING: OpenProcess failed with error 5 (Access is denied)
process ID = 0x00000EE8
thread count = 1
parent process ID = 0x00000EDC
Priority Base = 8
WARNING: CreateToolhelp32Snapshot (of modules) failed with error 5 (
THREAD ID = 0x00000EEC
base priority = 8
delta priority = 0
```

If you can see the message "OpenProcess failed with error 5 (Access is denied)", then you will know for sure that the casino user can't find out anything incriminating about Bring. Your security checks are now complete. Close Notepad, solitaire and the "cmd.exe" shortcut. Delete Gaze from your desktop and empty the recycle bin. Terminate the Bring process. Reboot the virtual computer.

Step 8 — **Install the ↓casino software**

You can now install the casino software. On the virtual computer, log in as the casino user. Create a new folder on the desktop called "poker". It's a good idea to install all your casino software in a single folder.

The casino user currently has restricted privileges. Some casinos insist that you install their software when logged in as a user with administrative privileges. Some casinos also insist that you run their software when logged in as a user with administrative privileges. Preferably, we would like to install and run the software while logged in as a user has restricted privileges.

Here is how to set up a typical casino's poker software.

Still logged in as the casino user, go to the casino web site. Download their software directly into the poker folder you created moments ago. Go into the folder and double click on the software. It will begin installing. When the install asks where you would like to install the software, choose the "poker" folder you created moments ago. Don't let the install the use the default folder.

If the installation completes without complaints, all is well. However, if the installation complaints that it needs administrative privileges, this is what you must do:

Log out from the casino user's account and log in to the admin user's account. Open the "User Accounts" window just as you did before. Click on the casino user act and then click on "Change the account type". Click on "Computer

Administrator" and then click the "Change Account Type" button. Log out from the admin user's account and log back into the casino user's account.

Try to install the software again. This time it should install without complaints. Now log out of the casino user's account and go back to the admin user account. Change the casino user's account type back to "limited". Log out of the admin user account and log back into the casino user's account.

Now try running the software. If the software complaints that it needs administrative privileges to run, change the casino user's account to "Computer Administrator" just as you did above. It's unfortunate that you have to run the software with administrative privileges, but it's not the end of the world.

Step 9 — Change your video **↓**settings

OpenHoldem is quite fussy about which desktop theme you are using. Currently, only the \(\pm\)Windows XP default style and the \(\pm\)Windows 2000 \(\pm\)classic style are supported. You must change your desktop theme to one of these styles on both the real computer and the virtual computer. Click "Start - Control Panel - (Appearances and Themes) - Change the computer's theme" to do this.

Step 10 — Start playing poker

On the virtual computer, start Bring by using the "cmd.exe" shortcut. Start the casino software and choose the table. Make sure the table's window is not obscured by any other Windows.

Now go to your real computer, start Bring and try to get the table to appear in a window. If everything has been set up properly, you should be able to control the table from your real computer.

Now start OpenHoldem. Press OpenHoldem's "Green Circle Button" button and you're ready to make money!

Step 11 — How to play in the future

Every time you use OpenHoldem to play poker for you, you must follow this simple checklist.

- 1. Boot up your real computer. Start OpenHoldem.
- 2. Boot up your virtual computer and log in as the casino user.
- 3. On the virtual computer, use the "cmd.exe" shortcut to run Bring.

- 4. Start the casino software. Select a table. Make sure this window isn't obscured by any other windows.
- 5. On the real computer, start Bring. Find the window with the poker table.

Outdated Features and Incompatibilities between OpenHoldem and WinHoldem

The original main goal of the OpenHoldem development team was the attempt to create a WinHoldem-compatible botting-platform to support existing bots, DLLs and TableMaps. However there has been functionality such as "Teaming", that always has been a tabu for the OpenHoldem development team. Other functionality did never get implemented due to lack of documentation and practical relevance. And now that OpenHoldem is far more advanced and WinHoldem is practically dead, backward-compatibility is no longer such an important issue. Recently the development team revisits some features that look a bit mal-designed and tries to improve them carefully, trying to affect the enduser as little as possible. This chapter will tell you about such incompatibilities.

No Collusion aka "Teaming"

OpenHoldem does not support any kind of collusion. There is no support for WinHoldems WinTellem-server and card-sharing either because such activities are simply illegal in every country of the world, and — even more important — because we consider them unfair too.



TableMaps

Table Maps are in a different format due to hypothetical copyright-issues. For some time it was possible to convert a WinHoldem "profile" to an OpenHoldem "Table Map" with OpenScrape. But this is no longer of any practical relevance, as all WinHoldem-"profiles" are supposed to be outdated and there is no longer any maintenance going on. So finally we removed support for WinHoldem-tablemaps completely.

Formula-File-Format

The formula is saved in a single file instead of two. It uses the extension *.ohf instead of *.whf and *.whx. Old WH-formulae could be converted for some time, but as there was no longer any demand for this feature we were happy to simplify OpenHoldems code. OpenHoldem does still use a plain-text-format, so *.ohf-files can be edited using your favourite text-editor. or the built-in formula-editor.

ManualMode

ManualMode is not directly integrated into OpenHoldem, but became a standalone application. This did simplify the GUI of OpenHoldem extremely and made it far more easy to use.

f\$evrais and f\$evcall

Calculating expectation values is only one possible way (out of many) to implement a poker bot, but not necessary in general. Furthermore it proved to be cumbersome and unpopular. So the functions f\$evrais and f\$evcall from WinHoldems demo-bot have been degraded to user-defined-functions.

f\$play-Function

The function f\$play has been a quite messy concept to handle sitting in, sitting out and leaving the table. Some extra coution was necessary to avoid multiple button-clicks within a short time-frame on identical or outdated game-states. With OpenHoldem 4.0.0 f\$play got replaced by three indivudal functions: f\$sitin, f\$sitout and f\$leave, which get executed exactly once every 4 seconds. Furthermore we added a f\$close-function to click the [x] in the titlebar of a window if the user desires to leave a table.

f\$swag function

f\$swag (short for "stated wager") was WinHoldems way to define the betsize for No-Limit games. As online casinos interpret your input in at least three different ways some adjustment was necessary, including:

the desired amount to raise f\$srai (the part which exceeds the

bet of the aggressor)

f\$swag the final amount to be

entered into the software

the casino-specific

f\$swag_adjust adjustment between f\$srai

and f\$swag

the casino-specific way to swagtextmethod interpret swag, as defined

in the tablemap

sraimin the smallest possible raise sraimax the largest possible raise. sraiprev the difference between the

two largest unique wagers

All these computations had to be done by the user at the formula level. With OpenHoldem 4.0.0 this mess got replaced by a single function f\$betsize and OpenHoldem cares about all the rest automatically (depending on the swagtextmethod specified in your tablemap, but that is all work that's left). The supporting tablemap symbols got renamed.

Former TM-symbol New TM-symbol

swagselectionmethod betsizeselectionmethod

swagdeletionmethodmethod betsizedeletionmethodmethod swagconfirmationmethod betsizeconfirmationmethod swagtextmethod betsizeinterpretationmethod

Statistic-Symbols: "risk of ruin"

The symbols to calculate "risk of ruin" have never been implemented due to lack of documentation and lack of practical relevance. As of OpenHoldem 2.0.0 the risk-of-ruin-symbols are officially abolished and removed from the code-base. The original documentation for these symbols can be found here:

- WinHoldEm Help Formula Editor (http://www.winholdem.net/help/help-formula.html)
- WinHoldem-Forum: Risk Of Ruin (http://forum.winholdem.net/wbb/viewtopic.php?t=2120)

Outdated Symbols

OpenHoldem is now half a dozen years old. During these long period of usage it became noticeable that some symbols and features of WinHoldem were unnecessary (risk of ruin), some others unusable, some undocumented (versus2), and some even undesired.

and some even unde	esirea.	
Symbol	Explanation	Reason for removal
nfriendsdealt	1 if you are dealt, 0 otherwise (0-1)	OpenHoldem does not support any kind of collusion, so the friend-symbols are completely meaningless. Only nfriendsseated, nfriendsactive and nfriendsplaying got kept for some time for backward-compatibility, as they were quite convenient to use in the f\$sitin, f\$sitout and f\$leave-formulas. They did consider hero as the one and only "friend". The last friend symbols (with the same restricted meaning) got removed from the code-base in OpenHoldem 4.0.0. We simply don't want that sh**, not even for backward-compatibility to WinHoldem.
nfriendsblind	1 if you are in a blind, otherwise (0-1)	, 0

nfriendsblind	1 if you are in a blind, 0 otherwise (0-1)
friendsseatedbits	bits 9-0: 1=seated 0=unseated, you only
friendsactivebits	bits 9-0: 1=active 0=inactive, you only
friendsdealtbits	bits 9-0: 1=dealt 0=notdealt, you only
	bits 9-0: 1=playing

friendsplayingbits 0=notplaying, you only bits 9-0: 1=blind friendsblindbits 0=notblind, you only Autoplayer delay in Some tablemap symbols have milliseconds between been exposed to the scripting level without necessity. They swag keystrokes and swagdelay button click as defined got removed in OpenHoldem in Preferences 2.2.0 and are now only (formerly TableMap) accessible internally. Autoplayer delay in milliseconds between alli slider jam and allidelay button click as defined in Preferences (formerly TableMap); OpenHoldem will now automatically chose an appropriate method 1. try to click allin (or max), optionally followed by clicking raise (if allinconfirmationmethod is non-zero). used to detrmine which 2. try to use the slider if it allinmethod method to use for allin exists in the tablemap. 3. finally try to swag the balance. This new behaviour tries the most fail-safe and most

> in addition reacts flexibly on the options available. The list symbols below seem to be mis-conceptions by WinHoldems author and — to

our best knowledge — have

human-like methods first and

true if your hand is in

islistcall list 0 never been used in any serious poker bot. They finally got removed from the code base in version 4.0.0 of OpenHoldem.

true if your hand is in islistrais

list 1

true if your hand is in islistalli

list 7

true if the call list is isemptylistcall

empty

true if the rais list is isemptylistrais

empty

true if the alli list is isemptylistalli

empty

highest list number in which your hand is

listed

lowest list number in nlistmin which your hand is

listed

Removing formula symbol "handnumber" for various reasons from the formula level:

the site hand number if

available

• it may contain characters and therefore doesn't fit into a variable of type double.

there is no need for it at

the formula level.

• its length may exceed the precision of double.

"Handnumber" gets now used only internally to trigger handresets.

handnumber

nlistmax

isfiveofakind	true when you have a five of a kind	Removed, as never used in Texas Hold'em
fiveofakind	0xff000000 (pokerval constant)	
isppro	true if you're connected to a ppro server	Removed together with the support for WinHoldem PokerPro-server, which did not get used and did no longer work due to undocumented changes in the network-protocal.
site	0=user/ppro 1=scraped	
bankroll	the user defined, real world bankroll	Removed, as these symbols have rarely been used and should better get handled by a user-defined formula.
rake	percentage amount added/subtracted to/from the pot	
defcon	be used in standard	Defcon looked like a WinHoldem-misconception which has been very rarely (if ever) used. People who need it can create a UDF for that, but it clearly is not necessary for the majority of users and does not deserve symbols and an extra settings dialog.
isdefmode	true when defcon is at max	

isaggmode true when defcon is at min number of CPU clocks Useless information and since the last screen clocks always a fraction of a second scrape number of CPU clocks Simply never used nclockspersecond per second synonym for ncps nclockspersecond Removed, because duplicate functionality and lack of short for nflopc ncommoncardsknow verbosity are bad softwareengineering short for betround br short for ncps nclockspersecond short for oppdealt nopponentsdealt true if OpenHoldem is Removed, because there is no need for that info at the attached to a Bring isbring client window formula-level true if you're in manual ismanual mode, false otherwise Getting rid of the symbol handrank and the option symbols->Value of Handrank. We have handrank 169, handrank1000, handrank1326, one of the handrankhandrank2625 and handrankp symbols based on the handrank at the formula level. So the option specified in user can clearly specify what preferences he wants. But these symbols can hardly get used interchangeably, so this option was only confusing and superfluos.

randomround1 randomround4	random number in the range [0.0001.000) for round 1 - 4. Value is calculated only once in that round.	Removed the symbols randomround1randomround4 from the codebase, because we have randomround for the current betting round and additionally randomhand, randomheartbeat and random, which gets calculated each time new. This simplified the code and looks more clean. The symbols "callshort" and "raisshort" got removed from
callshort	total amount that will be added to the pot if all players call	the code-base because they got designed for Fixed-Limit no- foldem Hold'em only. Better use a function to estimate future pot-sizes.
raisshort	callshort + bet * nplayersplaying	
seatposition	your seat position relative to the dealer	Removing the symbol seatposition which counted both active and inactive players and even empty chairs and was of no real use for practical play.
elapsed1970	time in seconds since 1970-01-01 00:00:00 GMT (Thursday)	Removing the symbol elapsed1970 because there was no real use for it.
ncommoncardspresent	number of common cards present (normal or highlighted)	Removing the symbol "ncommoncardspresent", because it was never implemented correctly, but always had the same value as "ncommoncardsknown". Furthermore its value would only differ at some casinos and at showdown (highlighted cards / bad scrapes), but this

point of time is pretty meaningless for both OHscript and OpenPPL, which evaluate mainly on the users turn, whereas DLLers still have access to all info.

1.	no callers or blinds
	only

2. called Pot — 1 bet to call

3. raised Back — 1 more bet to call because someone behind you raised after you've already bet/called/raised

Removed the symbols ac_pf_bets, because it was for fixed limit only, worked only on the first orbit preflop, because that task can be done with other symbols and finally we have the OpenPPL-library, so there is really no longer any need for that.

4. raised Pot — 2 bets to call

5. reraised Pot — 3+ bets to call

Only valid when betround == 1 which chair was aggressor (might be

from previous round)

Removed, as it was duplicate functionality to raischair

nplayersblind

ac_aggressor

ac_pf_bets

Removed with the introduction number of players blind of of the more practical (including you) (0-10) symbols smallbindchair and bigblindchair

nopponentsblind

number of opponents blind (not including you) (0-9)

playersblindbits

blind

bits 9-0: 1=blind 0=not

bits 9-0: 1=blind 0=not

opponentsblindbits blind

bblindbits bits 9-0: 1=big blind

0=not big blind

These symbols report the total number of possible river endings for the opponent (ron\$)

and the user (run\$). A

ron\$ / run\$ symbols value of zero means that type of poker hand

is not possible. Any non-zero value means that type of poker hand

will be seen that many

times

river opponent

ron\$royfl number : possible royal

flush

river opponent

ron\$strfl number : possible

straight flush

river opponent

ron\$4kind number: possible four

of a kind

river opponent

ron\$fullh number : possible full

house

ron\$flush river opponent

number: possible flush

river opponent

ron\$strai number : possible

straight

river opponent

ron\$3kind number: possible three

of a kind

river opponent

Removed the run\$/ron\$symbols, because they looked like a mis-conception, nobody used them, some numbers were wrong and the code was

unfixable (1000s of undocumented number)

ron\$2pair number : possible two

pair

river opponent

ron\$1pair number : possible one

pair

river opponent

ron\$hcard number : possible high

card

river opponent

ron\$total number : sum of all

possible river endings

the maximum possible

ron\$pokervalmax pokerval for the

opponent

opponent chances of

ron\$prnuts hitting the nuts on or

before the river

opponent chances of

ron\$prbest hitting pokervalmax on

or before the river

total number of cpu

ron\$clocks clocks used to calculate

the ron\$ symbols

similar like the

run\$... ron\$symbols above,

just for the user

PrWin-functions

Some supporting functions and symbols for the PrWin-simulations got renamed in OpenHoldem 4.0.0 to achieve more verbosity and better formula-grouping in the editor.

Former Name New Name

f\$P f\$prwin_number_of_opponents NIT f\$prwin_number_of_iterations

f\$topclip f\$prwin_topclip f\$mustplay f\$prwin_mustplay f\$willplay f\$prwin_willplay f\$wontplay f\$prwin_wontplay

Contributing to the Manual

This document is based on the old manual for OpenHoldem 2.0, lots of former Wiki-entries and lots of forum-posts at http://www.maxinmontreal.com/forums/. Many people contributed to it, but as you can see this manual is neither complete nor perfect. Lets blame the people who put it together for everything; -) If you have some corrections or additions and want to help us to improve this manual then feel welcome to get in contact with the development team.

Technical reference for developers

- the sources of this manual can be checked out from https://github.com/OpenHoldem/openholdembot/tree/master/Documentation
- to edit the sources we use LyX version 2.0.3. LyX is a graphical frontent for the mathematical type-setting-system LaTeX, but don't get frightened: it is simply a what-you-*mean*-is-what-you-get editor. LyX is available at: www.lyx.org/.
- LyX usually creates PDF, PS or DVI-files. However we export everything to HTML (File -> Export -> HTML).
- Finally we use MicroSofts HTML-help-workshop to create a "compiled HTML" help-file (*.CHM). The HTML-help-workshop is available here: http://www.microsoft.com/en-us/download/details.aspx?id=21138