Houdini 3 Chess Engine

User's Guide

Houdini 3

The Leading Chess Engine

by Robert Houdart

Houdini is a state-of-the-art chess engine combining outstanding positional evaluation with the most sophisticated search algorithm.

The name Houdini was chosen because of the engine's positional style, its tenacity in difficult positions and its ability to defend stubbornly and escape with a draw – sometimes by the narrowest of margins. On the other hand Houdini will often use razor-sharp tactics to deny its opponents escape routes when it has the better position.

Houdini 3 Chess Engine

© 2012 Cruxis

Printed: October 2012

Table of Contents

Part I

1 Introduction

Houdini is a state-of-the-art chess engine for Windows combining outstanding positional evaluation with the most sophisticated search algorithm. The name Houdini was chosen because of the engine's positional style, its tenacity in difficult positions and its ability to defend stubbornly and escape with a draw – sometimes by the narrowest of margins. On the other hand Houdini will often use razor-sharp tactics to deny its opponents escape routes when it has the better position.

Houdini is available in a Standard version and Professional version for power users with high-end hardware.

Houdini is written for Windows and will run on any not too ancient Windows version. On Linux you can run Houdini using Wine without any significant performance loss.

Since December 2010 Houdini has been leading all the major Computer Chess rating lists and is widely considered to be the strongest chess engine on the planet. From numerous interviews it appears that many top Grand Masters including the current chess World Champion Viswanathan Anand use and appreciate Houdini.

For recent information and updates please consult the Houdini home page at
http://www.cruxis.com/chess/houdini.htm

Feel free to contact the Houdini team with suggestions, problem reports or questions at the following e-mail address:
houdini@cruxis.be

1.1 Version History

Houdini 3

Released October 15, 2012.

Houdini 3 contains many evaluation and search improvements in all phases of the game and is about **50 Elo stronger** than its predecessor Houdini 2 (which at the time of writing was still unmatched by any other engine). The opening improvements relate mostly to piece mobility and space management and are most convincingly demonstrated by the progress in Fischer Random Chess for which Houdini 3 has become about 75 Elo stronger. In the middle game Houdini 3 has significant enhancements for recognizing pieces with limited mobility and in king-side safety. Houdini 3 will seek deeper in end games and solve more positions than before.

In the new **Tactical Mode** Houdini 3 will prefer tactical instead of positional solutions. In tactical test suites the Tactical Mode will find more solutions and provide significantly faster solution times, often by a factor of 3 to 10.

The Accelerated Principal Variation Search or "**Smart Fail-High**" is especially useful in very deep analysis when a different move becomes best at very high search depth. Houdini 3 will apply an automatic depth reduction that often speeds up finding the Principal Variation by a factor of 5 to 10.

Besides Nalimov and Gaviota End Game Table Bases, Houdini 3 now also supports **Scorpio bitbases**. These bitbases are loaded in memory when the program starts (requiring about 300 MB of memory) and are then readily available to the engine.

Hash usage has been optimized, improving back-tracking analysis. Houdini 3 Pro will now support hash tables up to 256 GB.

The **engine evaluations** have been carefully recalibrated so that +1.00 pawn advantage gives a 80% chance of winning the game against an equal opponent at blitz time control. At +2.00 the engine will win 95% of the time, and at +3.00 about 99% of the time. If the advantage is +0.50, expect to win nearly 50% of the time.

Fun fact: Over 10 million chess games were played for the development and tuning of Houdini 3!

Houdini 2.0c

Released November 20, 2011.

Corrections
- Nalimov EGTB probing would not recognize KvK end game.
- Nalimov EGTB: Houdini would exit when a corrupt table base file was encountered.
- On some Windows systems the internal timer would overflow after approx. 30 minutes.

Improvements and New Features
- New MultiPV_cp option to limit multi-PV analysis to moves within a range of the best move.
- New FiftyMoveDistance option to make the 50-move rule kick in earlier.
- New UCI_Elo and UCI_LimitStrength options as UCI standard-compliant alternative to Strength option.
- Houdini now exits when it detects that the communication with the GUI is broken.

Houdini 2.0b

Released October 7, 2011.

Corrections
- Repetition in games would have Houdini evaluate the repeating position as 0.00.
- When specifying a fixed depth search, Houdini would wait for a "stop" command when analysis finished early.
- Omitting the "uci" command at start-up would produce erratic results in the first analysis.

Improvements and New Features
- Nalimov EGTB support.

Houdini 2.0

Released September 1, 2011.

Improved Analysis support: Save Hash to File, Load Hash from File, Never Clear Hash
The complete hash table can be saved to a disk file and reloaded later for continuing the analysis.
Houdini 2 makes better use of the data in the hash table to restart instantly the analysis at the point where it was previously interrupted.
By saving the hash table in a disk file you can interrupt the analysis anytime. At a later date you can reload the hash table file in memory to continue the analysis as if no interruption had happened.
The new "Never Clear Hash" option will keep the hash data in memory even when the position changes or when starting a new game.

Position Learning
The "Position Learning" mode will automatically save analysis results in a learning database that will be reused in future analysis. This is a more convenient, but less powerful method of using previous analysis results than the manual Save Hash/Load Hash operations of the previous point.

Strength Limit feature
Houdini 2 can limit its skill level from 0 (beginner) to 100 (full strength).
The strength is limited through a combination of techniques: limiting the number of positions searched, purposely picking a non-optimal move, and ignoring table bases.
Up to what skill level can you beat Houdini?

Chess960 (Fischer Random Chess) Support
Houdini 2 supports Chess960 a.k.a. Fischer Random Chess games.

Mate Search
To speed up solving deep mates, you can fix a limit on the search depth during the mate search.

Houdini Pro version for high-end users with powerful hardware
For advanced servers with many cores on multiple sockets (usually with so-called NUMA architecture), a special Houdini Pro version is available. It significantly improves the scaling of Houdini up to 32 cores and makes use of up to 32 GB of hash memory.

Improved strength
Improved evaluation and search make Houdini 2 Standard about 25 Elo stronger than its predecessor Houdini 1.5a. On high-end hardware Houdini 2 Pro with its additional NUMA-related speed improvements is about 40 Elo stronger than Houdini 1.5ab-16.

Part II

2 Installing and Running Houdini

2.1 Installing Houdini

Houdini is written for Windows and will run on any not too ancient Windows version. On Linux you can run Houdini using Wine without any significant performance loss.

You should have received the download link for the Houdini Setup, together with your Customer Code and Serial Number.

Download and run the Houdini Setup package. During installation you will be asked for the Customer Code and Serial Number. The Houdini Setup will perform an internet Activation. Without activation Houdini will run at reduced strength.

The Houdini Setup will install the Houdini version that is appropriate for your 32-bit or 64-bit system. The Houdini engine contains optimized code for single-core up to 6 cores (Houdini Standard version) or up to 32 cores (Houdini Pro version). The 64-bit version is about 30% faster than the 32-bit version. On modern hardware the program will use the POPCNT instruction for another slight performance gain.
Below a table summarizing the single-core performance difference on two modern processors:

Intel Core i5-750 @ 2.67 GHz		
Version	**Node Speed**	**Relative Speed**
Houdini 3 32-bit	1620 kN/s	1.00
Houdini 3 32-bit + POPCNT	1720 kN/s	1.06
Houdini 3 64-bit	2150 kN/s	1.33
Houdini 3 64-bit + POPCNT	2300 kN/s	1.42

AMD Opteron 6128 @ 2 GHz		
Version	**Node Speed**	**Relative Speed**
Houdini 3 32-bit	925 kN/s	1.00
Houdini 3 32-bit + POPCNT	965 kN/s	1.04
Houdini 3 64-bit	1135 kN/s	1.23
Houdini 3 64-bit + POPCNT	1260 kN/s	1.36

The table demonstrates that an upgrade to 64-bit Windows is highly recommended if you're still running 32-bit Windows. The benefit from the "POPCNT" feature is also evident, Houdini 3 has significantly more evaluation terms than its predecessors and will obtain great benefit from modern Intel or AMD 64-bit processors with POPCNT.

2.2 Running Houdini in a graphical chess environment

Houdini is a so-called "UCI-compatible" chess engine that requires a graphical chess environment or chess "GUI" (Graphical User Interface) to run. This GUI will show you the chess board and allow you to play chess games or analyze positions.

For the best user experience of Houdini you'll need to install a graphical chess environment like the free Arena, SCID or Winboard, or the commercial Shredder, ChessBase or Aquarium.

After installing the chess environment, you can install Houdini as a UCI engine:
- Arena: use the "Engines", "Install New Engine" menu.
- Shredder Classic GUI: use the "Extras", "Engines", "Install Engine" menu.
- Fritz/ChessBase GUI: use the "Engine" tab, "Create UCI Engine" button.
- Aquarium: use the "Add" button on the "Engines" tab.

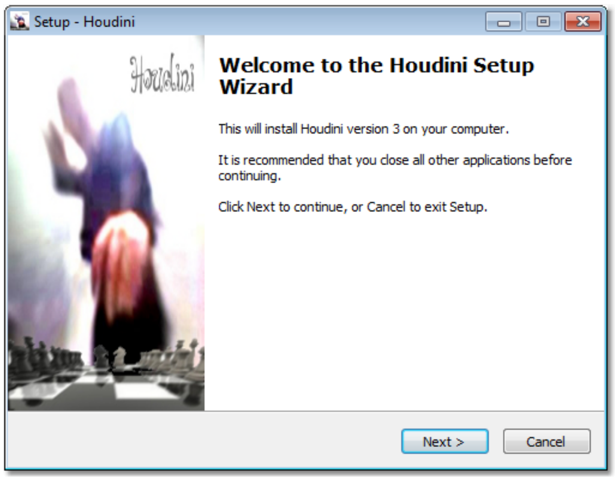
When a file selection dialog opens, go to the "C:\Program Files\Houdini 3 Chess" folder and select the "Houdini_3_w32.exe" or "Houdini_3_x64.exe" executable (whichever is present).

The next chapter contains a step-by-step installation of Houdini and Arena.

2.3 Installation with Arena 3.0

1) Download Houdini Setup from the download link you received by e-mail.

2) Run the Houdini Setup.
After selecting the installation language you'll see the following welcome screen:



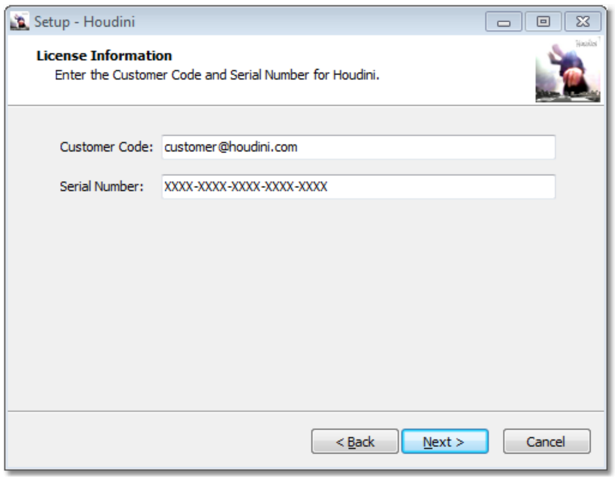
Welcome to the Houdini Setup Wizard

This will install Houdini version 3 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

After approving the license, you'll be asked for your Customer Code and Serial Number. Simply copy/paste the information you received by e-mail.



License Information

Enter the Customer Code and Serial Number for Houdini.

Customer Code: customer@houdini.com

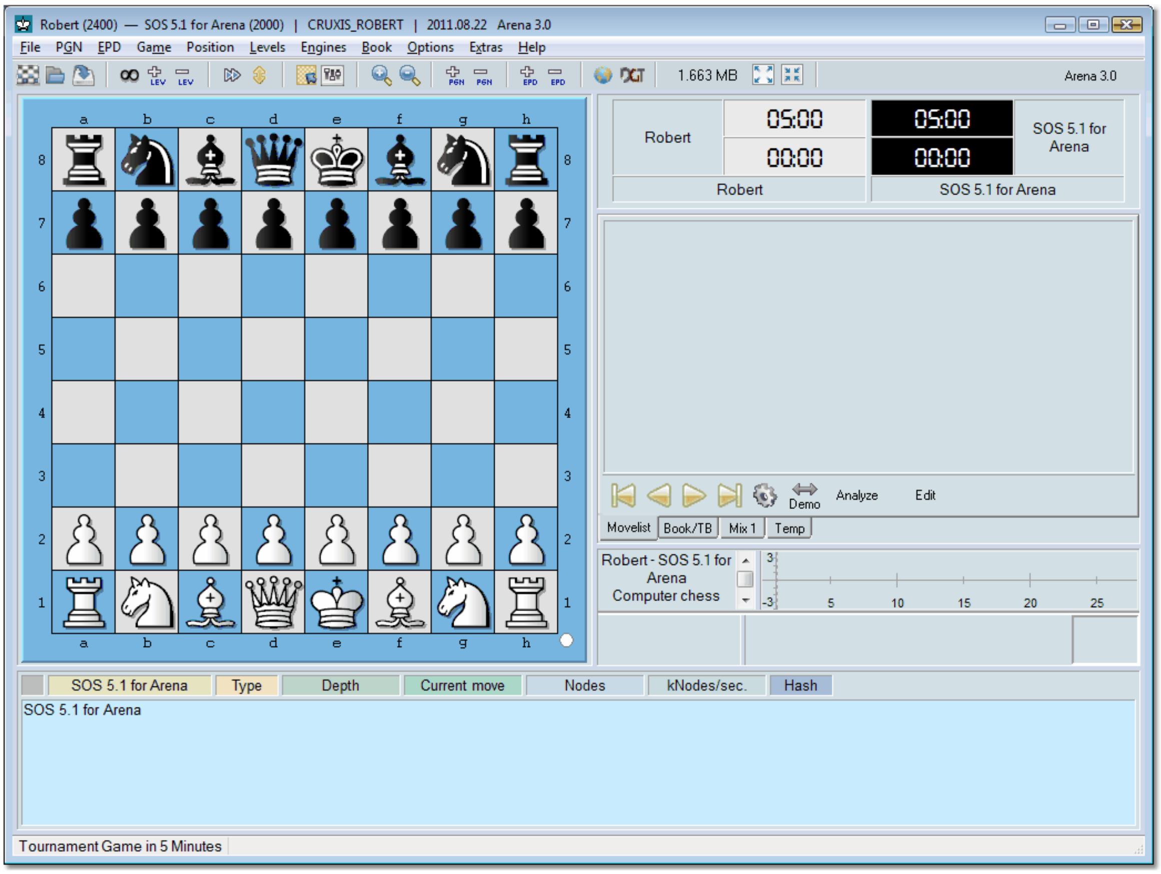
Serial Number: XXXX-XXXX-XXXX-XXXX-XXXX

Finish the installation right to the end.

3) Download the Arena Setup from http://www.playwitharena.com. Go to the Downloads page and select the "Arena Setup" package.

4) Run the Arena Setup
All the default installation options are OK, just click the "Next" button.
At the end of the set-up you'll automatically run Arena.

5) Run Arena
After selecting your language, you'll find the main screen of Arena.

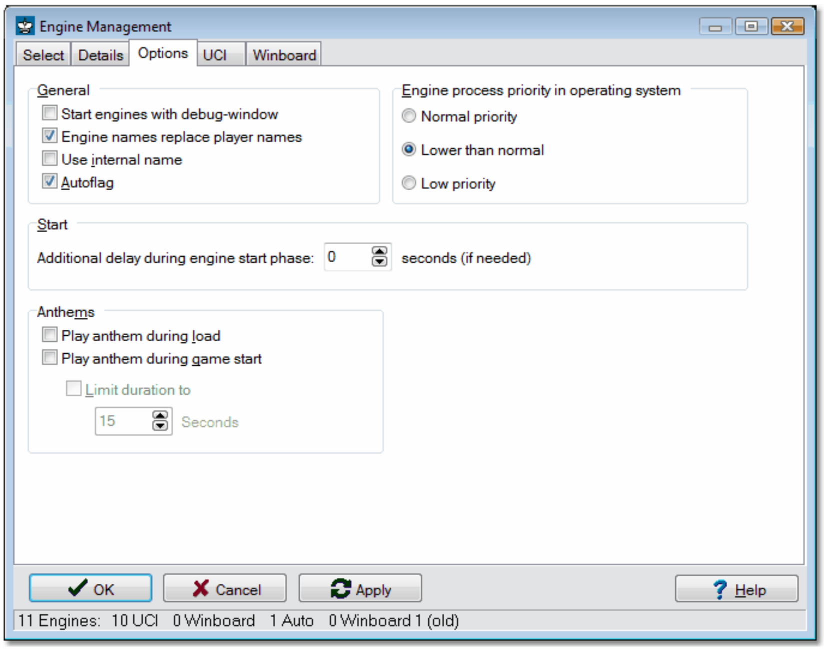

Robert (2400) — SOS 5.1 for Arena (2000) | CRUXIS_ROBERT | 2011.08.22 Arena 3.0
File PGN EPD Game Position Levels Engines Book Options Extras Help
1.663 MB
Arena 3.0
Robert
05:00
00:00
05:00
00:00
SOS 5.1 for Arena
Robert
SOS 5.1 for Arena
Demo
Analyze
Edit
Movelist
Book/TB
Mix 1
Temp
Robert - SOS 5.1 for Arena Computer chess
SOS 5.1 for Arena
Type
Depth
Current move
Nodes
kNodes/sec.
Hash
SOS 5.1 for Arena
Tournament Game in 5 Minutes

6) Install Houdini as a UCI engine
Click on the "Engines" menu, select "Install New Engine...".
A file selection dialog opens, go the "C:\Program Files\Houdini 3 Chess" folder and select the Houdini_3_w32.exe or Houdini_3_x64.exe executable.
Arena will then offer the choice between "UCI" and "Winboard", simply click OK to keep the default UCI.
Arena will start the Houdini engine.

7) Congratulations, you've successfully installed Houdini in Arena!
You can now play games or run analyses.

8) Some further recommended options:

A) Click on the "Engines" menu, select "Manage...", the Engine Management window pops up.
Go to the "Options" tab and select the "Lower than Normal" option in the "Engine process priority in operating system" group.


Engine Management

Select | Details | Options | UCI | Winboard

General
- [ ] Start engines with debug-window
- [✓] Engine names replace player names
- [ ] Use internal name
- [✓] Autoflag

Engine process priority in operating system
- Normal priority [✓ Lower than normal] Low priority

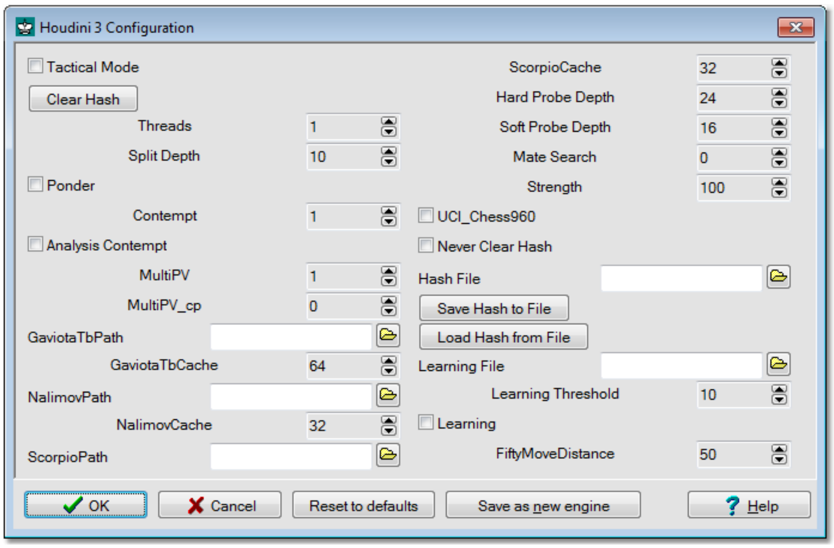
Start
Additional delay during engine start phase: 0 seconds (if needed)

Anthems
- [ ] Play anthem during load
- [ ] Play anthem during game start
- [ ] Limit duration to 15 Seconds

OK | Cancel | Apply | Help

11 Engines: 10 UCI 0 Winboard 1 Auto 0 Winboard 1 (old)

B) Adapt the Strength of Houdini to match your skill level.
Right-click on the bottom part of the Arena window and select the "Configure Houdini_20" (or press CTRL-1) option.
The configuration window will pop-up:


Houdini 3 Configuration

[ ] Tactical Mode		ScorpioCache	32
Clear Hash		Hard Probe Depth	24
Threads	1	Soft Probe Depth	16
Split Depth	10	Mate Search	0
[ ] Ponder		Strength	100
Contempt	1	[ ] UCI_Chess960	
[ ] Analysis Contempt		[ ] Never Clear Hash	
MultiPV	1	Hash File	
MultiPV_cp	0	Save Hash to File	
GaviotaTbPath		Load Hash from File	
GaviotaTbCache	64	Learning File	
NalimovPath		Learning Threshold	10
NalimovCache	32	[ ] Learning	
ScorpioPath		FiftyMoveDistance	50

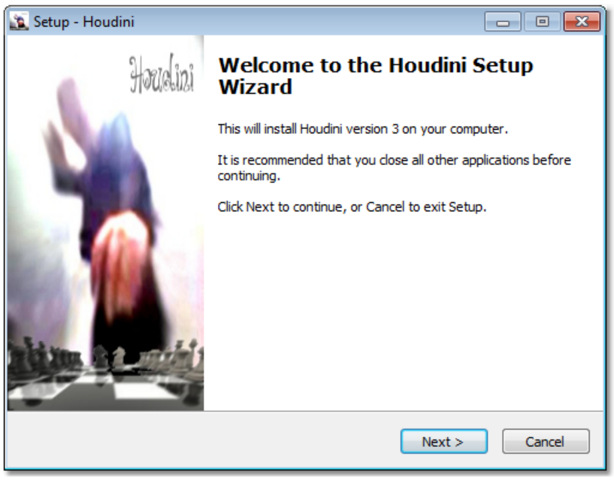
OK | Cancel | Reset to defaults | Save as new engine | Help

By default the Strength option is at 100 (full strength), you can try to adapt this value so that you get competitive games.
What is the highest strength level that you can consistently beat?
P.S. Don't forget to switch the Strength back to 100 when analyzing positions.

2.4 Installation with Fritz/ChessBase GUI

1) Download Houdini Setup from the download link you received by e-mail.

2) Run the Houdini Setup **outside** the Fritz/ChessBase GUI.
After selecting the installation language you'll see the following welcome screen:



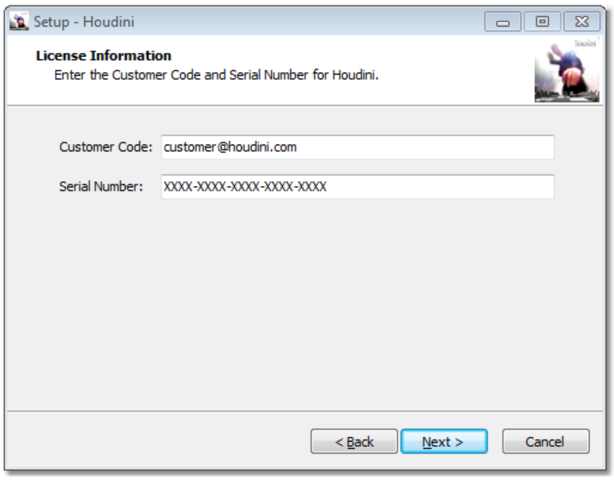
Welcome to the Houdini Setup Wizard

This will install Houdini version 3 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

After approving the license, you'll be asked for your Customer Code and Serial Number. Simply copy/paste the information you received by e-mail.

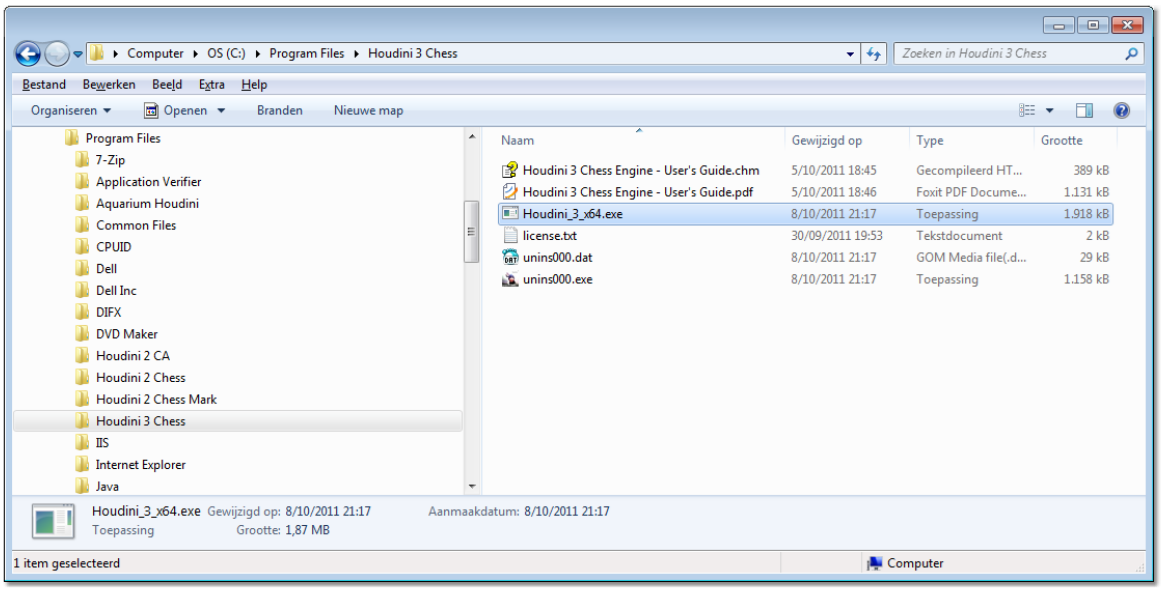


License Information
Enter the Customer Code and Serial Number for Houdini.

Customer Code: customer@houdini.com
Serial Number: XXXX-XXXX-XXXX-XXXX-XXXX

Finish the installation right to the end.

3) All the Houdini files have now been installed on your hard disk in the folder "C:\Program Files\Houdini 3 Chess".
You should find them as in the following image (Houdini Standard on 64-bit Windows):



Computer ▸ OS (C:) ▸ Program Files ▸ Houdini 3 Chess

Zoeken in Houdini 3 Chess

Bestand Bewerken Beeld Extra Help

Organiseren ▾ Openen ▾ Branden Nieuwe map

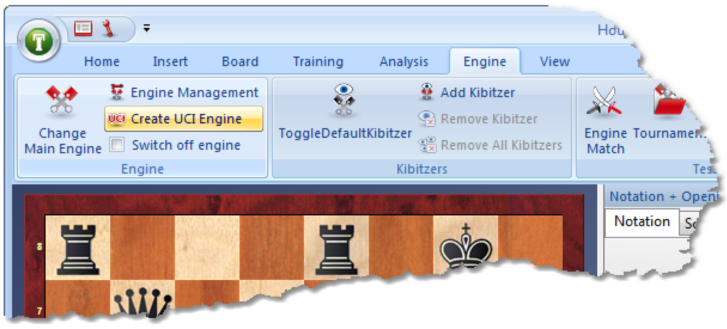
Program Files
- 7-Zip
- Application Verifier
- Aquarium Houdini
- Common Files
- CPUID
- Dell
- Dell Inc
- DIFX
- DVD Maker
- Houdini 2 CA
- Houdini 2 Chess
- Houdini 2 Chess Mark
- Houdini 3 Chess
- IIS
- Internet Explorer
- Java

Naam	Gewijzigd op	Type	Grootte
Houdini 3 Chess Engine - User's Guide.chm	5/10/2011 18:45	Gecompileerd HT...	389 kB
Houdini 3 Chess Engine - User's Guide.pdf	5/10/2011 18:46	Foxit PDF Docume...	1.131 kB
Houdini_3_x64.exe	8/10/2011 21:17	Toepassing	1.918 kB
license.txt	30/09/2011 19:53	Tekstdocument	2 kB
unins000.dat	8/10/2011 21:17	GOM Media file(.d...	29 kB
unins000.exe	8/10/2011 21:17	Toepassing	1.158 kB

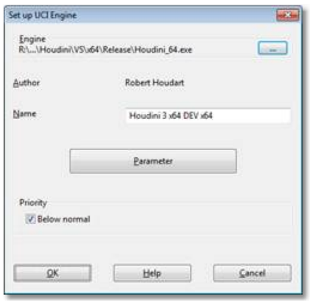
Houdini_3_x64.exe Gewijzigd op: 8/10/2011 21:17 Aanmaakdatum: 8/10/2011 21:17
Toepassing Grootte: 1,87 MB

1 item geselecteerd Computer

4) Run the Fritz/ChessBase GUI, select the "Engines" tab and click the "Create UCI Engine" button.



The following "Set up UCI engine" window will open:



Set up UCI Engine

Engine
R:\...\Houdini\VS\x64\Release\Houdini_64.exe

Author Robert Houdart

Name Houdini 3 x64 DEV x64

Parameter

Priority
[✓] Below normal

OK Help Cancel

Click on the "…" button in the upper right.
Select the engine file "Houdini_3_x64.exe" from the folder "C:\Program Files\Houdini 3 Chess".
Check the "Priority Below normal" box to keep the GUI responsive when the engine is running, then click OK.

Congratulations, you've now completed the Houdini installation in the Fritz/ChessBase GUI!

2.5 Installation on Linux using Wine/Wine64

To run Houdini on Linux you'll need to install Wine or Wine64.
Below some installation experiences and tips kindly shared by Houdini users. If you have additional experience - positive or negative - that could benefit other Houdini users, please let us know by e-mail and we'll add it in this manual.

Ubuntu 12.04

Success. I installed the 64-bit wine and then made another mistake and installed Houdini-3 again and once again got the 32-bit version. My mistake? The .wine directory on my system was already created under wine 32-bit. So ... removing that directory and THEN installing Houdini-3 results in a working 64-bit Houdini installation.

Arena 3.0 installed under 64-bit wine (running on Ubuntu 12.04 64-bit) is able to make use of the 64-bit Houdini 3 .exe file just fine. For those that like to run engine-vs-engine matches on their Linux computers, this will be good news. Until today I had not figured out how to install 64-bit wine and thus Arena was always limited to running 32-bit engines for me. No longer ...

The key steps for me in getting this to run on Ubuntu 12.04 were these:

1. To install wine, run the "Ubuntu Software Center" application.

2. In the upper right search bar, enter "wine" (minus the quotes of course)

3. Here was the real revelation for me -- click on the link shown as "show 25 technical items" at the bottom of the Ubuntu Software Center window. This reveals
Microsoft Windows Compatibility Layer (64-bit support). Click on that to install 64-bit wine. If you do not do this and just click on the "normal" wine, you get a 32-bit wine installed despite the fact that you are running Ubuntu 12.04 64-bit as an OS!

4. If you have a .wine directory already in your home directory, rename it to something else; I chose to rename it to: .wine-32bit   . This is crucial also since wine "knows" this is a 32-bit wine and the Houdini 3 installation step about to come will install 32-bit Houdini if you do not take this step.

5. Open a terminal window and execute: wine Houdini-3-Setup-b833c5.exe . This will create a new .wine directory in your home directory and this will be suitable for using 64-bit Windows applications.
Follow the Houdini 3 installation prompts and it will install a 64-bit Houdini-3 engine for you to work with. I use SCID to run Houdini 3 but Arena 3.0 works very well also.

Ubuntu 11.10

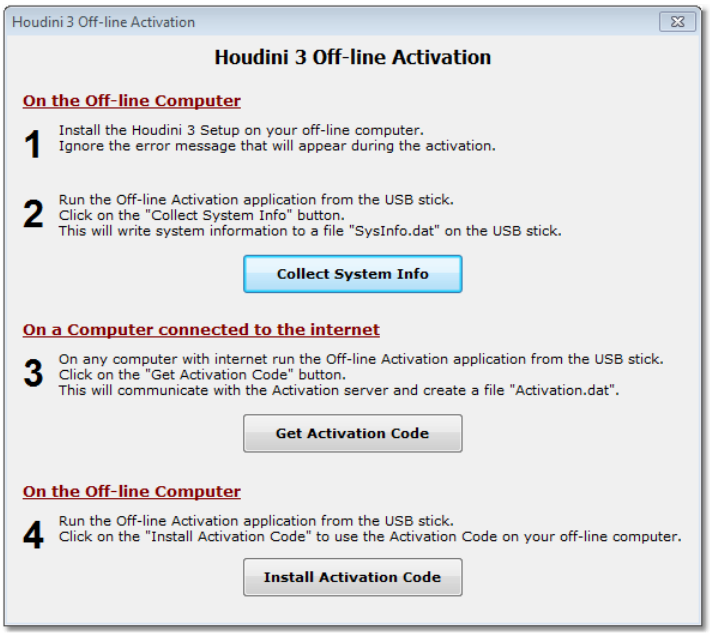
I found this script which did a parallel 32-bit + 64-bit install on Ubuntu 11.10, which got it working for me:
http://www.jesseo.com/chess/64-bit-wine-ubuntu.sh

2.6 Off-line Activation

Houdini 3 requires a product activation via internet.
If you want to install Houdini 3 on a computer without internet access, you can use the Off-line Activation module.

You will require any other computer with internet access, and a USB memory stick to transport files to and from the off-line computer. The process boils down to clicking 3 times on a button on the two computers and shouldn't take more than a couple of minutes. Note that NOTHING will be installed or modified on the computer with the internet access, it is only used to perform the communication with the activation server. The activation data is saved on the USB stick in two binary data files.

Below a step-by-step manual to perform the installation of Houdini 3 on a computer without internet access.



Houdini 3 Off-line Activation

Houdini 3 Off-line Activation

On the Off-line Computer

1 Install the Houdini 3 Setup on your off-line computer.
Ignore the error message that will appear during the activation.

2 Run the Off-line Activation application from the USB stick.
Click on the "Collect System Info" button.
This will write system information to a file "SysInfo.dat" on the USB stick.

Collect System Info

On a Computer connected to the internet

3 On any computer with internet run the Off-line Activation application from the USB stick.
Click on the "Get Activation Code" button.
This will communicate with the Activation server and create a file "Activation.dat".

Get Activation Code

On the Off-line Computer

4 Run the Off-line Activation application from the USB stick.
Click on the "Install Activation Code" to use the Activation Code on your off-line computer.

Install Activation Code

0. Prepare a USB memory stick on the computer with internet access

Download the Houdini Setup and copy to the USB memory stick.
Download the Off-line Activation Module and copy to the USB memory stick.
Write down your Customer Code and Serial Number, or copy the values to a text file on your USB stick.

1. Run the Houdini Setup on the off-line computer

Run the Houdini Setup from the USB memory stick. Enter your Customer Code and Serial Number when requested.
During production activation you'll get the error "Product Activation Failed" because of the missing internet connection, you can simply ignore it.

2. Run the Off-line Activation Module on the off-line computer

Run the Off-line Activation Module from the USB memory stick.

Click on the "**Collect System Info**" button.
A "SysInfo.dat" file will be written to the USB memory stick.

3. Run the Off-line Activation Module on the computer with internet access

Run the Off-line Activation Module from the USB memory stick.
Click on the "**Get Activation Code**" button.
The Activation Server will be contacted, and a "Activation.dat" file will be written to the USB memory stick.

4. Run the Off-line Activation Module on the off-line computer

Run the Off-line Activation Module from the USB memory stick.
Click on the "**Install Activation Code**" button.
The Activation code will be registered and the Houdini 3 engine files will be updated.

2.7 Some Frequently Asked Questions

Q: Sometimes when Houdini runs my computer appears to be frozen and goes very slow. What can I do to improve the responsiveness of my computer while the engine runs?

There are several ways to keep your computer more responsive while Houdini is thinking:

- Run the engine at "Lower than normal" priority. Most chess GUI will provide this option, for example:
  - Arena has a "Lower than normal process priority" checkbox in the Engine Management Options.
  - Shredder GUI will automatically run engines at lower priority.
  - Fritz/ChessBase has a "Priority Below Normal" checkbox in the dialog for creating the UCI engine.
- If you want to use your computer for other purposes while analyzing chess positions, set the "Threads" configuration option to a value smaller than the the number of physical cores of your computer - for example, use 3 on a quad-core computer. This will leave one core available for other applications while the chess analysis runs in the background.

Q: What Hash Size do you recommend at various time controls?

For infinite analysis or long Time Control matches you should use the largest hash possible - typically about half of the total RAM memory of your computer. For example, on a system with 4 GB of memory you can use up to 2048 MB hash size, on a 12 GB RAM system you can use up to 8192 MB.
For shorter games, e.g. 3 to 5 minute games, it's better to use 256 MB or 512 MB as this will provide the best performance. For 16 min games 1024 or 2048 MB hash size should be fine.

If you know the average move time T (in seconds) and the average node speed of your hardware S (in kN/s), you can compute the optimal hash size with the formula: (T x S / 100) MB.

For example if you use a Time Control of 10 minutes for 40 moves repeating, the average move time T = 15 seconds. On hardware that produces about 2,000 kN/s the optimal hash size would then be approximately (15 x 2,000 / 100) = 300 MB, in other words 256 MB or 512 MB would be the recommended values.

Q: I'm running Houdini on a Core i7 CPU with hyper-threading. Would you recommend to use hyper-threading with Houdini?

The architecture of Houdini (and of chess engines in general) is not very well suited for hyper-threading; using more threads than physical cores will usually degrade the performance of the engine. Although the hyper-threads often produce a slightly higher node speed, the increased inefficiency of the parallel alpha-beta search more than offsets the speed gain obtained with the additional hyper-threads.
To give a practical example, it's more efficient to use 4 threads running at 2,000 kN/s each than 8 threads running at 1,100 kN/s each, although the latter situation produces a higher total node speed.
For this reason it's best to set the number of threads not higher than the number of physical cores of your hardware.

Q: The Fritz/ChessBase GUI often seems to freeze when I use Houdini. Furthermore the Fritz/ChessBase GUI doesn't memorize my Houdini settings, I select an option but next time I use the engine the value has been reverted. What is happening?

You're not the only one struggling with the sometimes confusing Fritz/ChessBase interface.
First note that chess engines are very CPU-intensive - when configured to use all the cores of the computer they may prevent the graphical environment from interacting normally with the user. To avoid this, it is recommended to run Houdini at a lower process priority in Windows.

We can suggest the following.

1. In the "Create UCI Engine" dialog, pick the Houdini executable and check the Priority "Below normal" box. This should make a big difference with respect to the ChessBase freezes.

2. The engine default properties are stored in a .uci file stored in your local Application Data folder. It's a plain text file that you can easily modify with Notepad. Open a Windows Explorer window, type `%appdata%` in the address bar and press Enter. In the folder that appears, descend into the ChessBase\Engines.UCI folder in which you will find the .uci files of all the installed engines. You can simply add options to the .uci file with Notepad, here's an example file:

```
[ENGINE]
Name=Houdini 3 x64
Author=Robert Houdart
Filename=C:\Program Files\Houdini 3\Houdini_3_x64.exe
Priority=below normal

[OPTIONS]
Threads=3
GaviotaTbPath=C:\egtb\gaviota
```

Q: I use the ChessBase Deep Position Analysis feature. I notice that Houdini's analysis went down risky lines without taking note of drawing/equalizing chances almost every move. Is that to be expected?

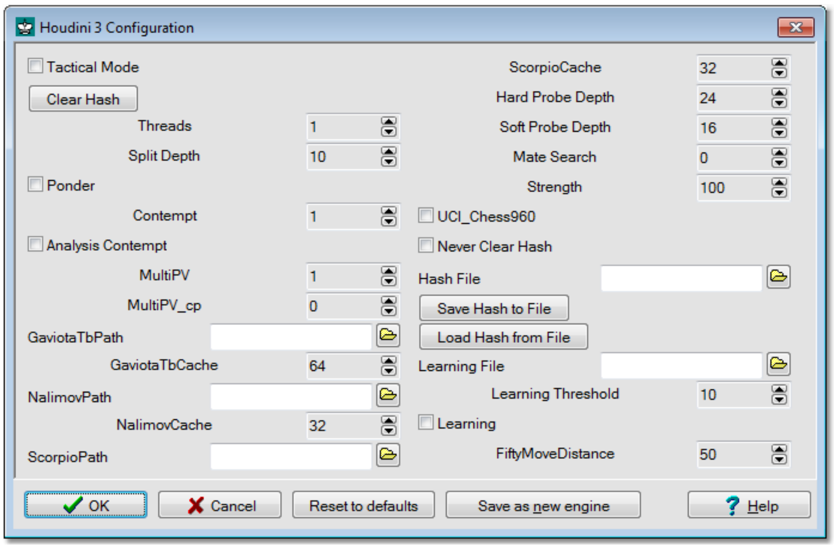
If you want Houdini to favor drawing lines, you should consider putting Houdini's "Contempt" value to 0.
ChessBase runs its deep position analysis by simulating game conditions. This means that Houdini thinks that it's in a game instead of an analysis, and activates the contempt (default value 1).
To avoid this, it's best to put the Contempt value to 0 when using the deep position analysis in ChessBase.

Part III

3 Houdini Configuration

Many engine options can be modified via the configuration window shown by the chess GUI.

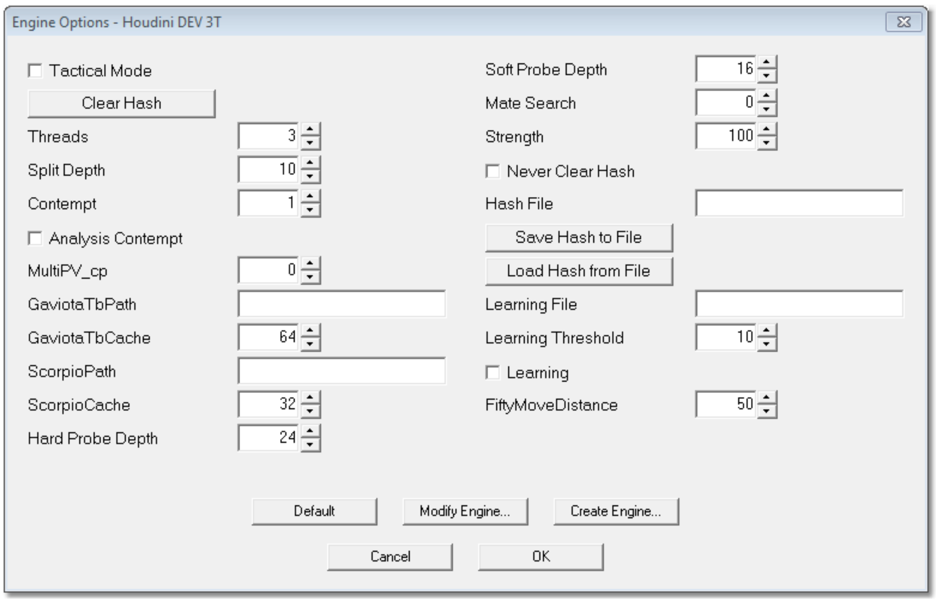
Below a screen shot of the configuration window in Arena 3.0.



Houdini 3 Configuration

- [ ] Tactical Mode
- Clear Hash
- Threads: 1
- Split Depth: 10
- [ ] Ponder
- Contempt: 1
- [ ] Analysis Contempt
- MultiPV: 1
- MultiPV_cp: 0
- GaviotaTbPath:
- GaviotaTbCache: 64
- NalimovPath:
- NalimovCache: 32
- ScorpioPath:
- ScorpioCache: 32
- Hard Probe Depth: 24
- Soft Probe Depth: 16
- Mate Search: 0
- Strength: 100
- [ ] UCI_Chess960
- [ ] Never Clear Hash
- Hash File:
- Save Hash to File
- Load Hash from File
- Learning File:
- Learning Threshold: 10
- [ ] Learning
- FiftyMoveDistance: 50

The same configuration window in the Shredder Classic 4 GUI:



Engine Options - Houdini DEV 3T

- [ ] Tactical Mode
- Clear Hash
- Threads: 3
- Split Depth: 10
- Contempt: 1
- [ ] Analysis Contempt
- MultiPV_cp: 0
- GaviotaTbPath:
- GaviotaTbCache: 64
- ScorpioPath:
- ScorpioCache: 32
- Hard Probe Depth: 24
- Soft Probe Depth: 16
- Mate Search: 0
- Strength: 100
- [ ] Never Clear Hash
- Hash File:
- Save Hash to File
- Load Hash from File
- Learning File:
- Learning Threshold: 10
- [ ] Learning
- FiftyMoveDistance: 50

Note that some GUIs will hide options that are handled elsewhere in its interface. For example, in the second screen shot above the options "Ponder", "MultiPV", "NalimovPath", "NalimovCache" and "UCI_Chess960" are missing from the configuration window.

3.1 Hash Memory

The so-called "Hash Memory" or "Transposition Table" is used by Houdini to store its analysis results.

Hash

Amount of hash table memory used by Houdini, in MB.
Default 128, min 4, max 1024 (32-bit) or 4096 (64-bit Standard) or 262144 (64-bit Pro).
The value is rounded down to a power of 2 (4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144 MB).

For infinite analysis or long time control matches you should use the largest hash that fits in the physical memory of your system. For example, on a system with 4 GB of memory you can use up to 2048 MB hash size.
For shorter games, for example 3 to 5 minute games, it's better to use 256 MB or 512 MB as this will provide the best performance. For 16 min games 1024 or 2048 MB hash size should be fine.

In the FAQ about Hash Size you'll find a formula to compute the optimal hash size for your hardware and time control.

Clear Hash (button)

Button to clear the Hash Memory.
If the Never Clear Hash option is enabled, this button doesn't do anything.

Never Clear Hash (checkbox)

This option prevents the Hash Memory from being cleared between successive games or positions belonging to different games.
Check this option also if you want to Load the Hash from disk file, otherwise your loaded Hash could be cleared by a subsequent `ucinewgame` or Clear Hash command.

3.2 Persistent Hash

Persistent Hash means that the in-memory hash table can be saved to a disk file and reloaded later for continuing the analysis.
The goal is to be able to interrupt a long position analysis anytime and save the hash table to a disk file. At a later date you can reload the hash file in memory and continue the analysis at the point where it was interrupted.

Correspondence chess players could profit from this feature by keeping one hash disk file per ongoing chess game. For that purpose the following procedure can be used.

To save a Hash file to disk:
- End the analysis
- Go to the options window, enter the name of the Hash File (e.g. C:\Chess\Game1.dat)
- Press the Save Hash to File button, and OK in the options window.

To load a Hash file from disk:
- Load the correspondence game
- Go to the options window, enter the name of the Hash File (e.g. C:\Chess\Game1.dat)
- Press the Load Hash from File button, and OK in the options window.

Hash File

File name for saving or loading the hash file with the Save Hash to File or Load Hash from File buttons.
A full file name is required, for example C:\Chess\Hash001.dat.
By default Houdini will use the hash.dat file in the "My Documents" folder of the current user.

Save Hash to File (button)

Save the current hash table to a disk file specified by the Hash File option.
Use the Save Hash to File button after ending the analysis of the position. Some GUIs (e.g. Shredder, Fritz) wait for sending the button command to the engine until you click OK in the engine options window.
The size of the file will be identical to the size of the hash memory, so this operation could take a while.
This feature can be used to interrupt and restart a deep analysis at any time.

Load Hash from File (button)

Load a previously saved hash file from disk.
Use the Load Hash from File button after loading the game or position, but before starting the analysis. Some GUIs (e.g. Shredder, Fritz) wait for sending the button command to the engine until you click OK in the engine options window.
The size of the hash memory will automatically be set to the size of the saved file.
Please make sure to check the Never Clear Hash option, as otherwise your loaded Hash could be cleared by a subsequent ucinewgame or Clear Hash command.

3.3 Position Learning

With Position Learning Houdini will remember advantageous principal variations it has analyzed by storing them in a learning disk file, for reuse in later games or analysis. The feature is completely transparent to the user, once Learning is enabled the data will be saved and loaded automatically.

The Position Learning is available in analysis and match play. But remember that it will only kick in for certain positions in which there is something to learn, i.e. when the score is significantly above the static evaluation of the position. In quiet, equal positions there's usually not much to learn.

Position Learning works best for positions where a good move exists that raises the score significantly above the statical evaluation. Learning is less useful and sometimes counterproductive for positions where many more or less equivalent moves exist and the goal is to avoid bad moves. To improve the efficiency of the learning the feature only kicks in when a certain positive scoring threshold is exceeded. By default this Learning Threshold is put at 10 centi-pawn (0.1 pawn).

Although the learning file grows very slowly, make sure not to allow the file to grow too large as the automatic loading may slow down the engine at the start of the analysis.

The impact of Position Learning can be illustrated as follows.
When running the Eigenmann Ending Test Suite with 20 seconds/move on a single-core laptop producing about 1,000 kN/s, Houdini finds 77 out of 100 solutions, the total solution time is 12:06. The test run produces a Learning File of about 27 kB.
Running the Test Suite a second time demonstrates the profit of the Learning File. Houdini now finds 80 solutions and the total solution time drops to 9:42.

Learning (checkbox)

When Learning is enabled the learning file will automatically be loaded at the start of the analysis, and further analysis will be appended to the file.
When Learning is enabled, automatically the **Never Clear Hash** option will be activated as clearing the hash would be counterproductive for the learning.

Learning File

File name for saving the learning data.
A full file name is required, for example C:\Chess\Learn001.dat.
By default Houdini will use the Learn.dat file in the "My Documents" folder of the current user.
The learning file will be updated at the start of a new game or when you close the engine or GUI.

Learning Threshold

Min 0, Max 200, Default 10.
The Learning Threshold is the minimum positive score of a principal variation that is saved.
When Learning Threshold is larger the learning file will grow slower, as fewer moves will satisfy the condition..

3.4 Cores and Threads Management

Threads

Maximum number of threads (cores) used by the analysis.
Default is hardware-dependent, min 1, max 6 (Standard) or 32 (Pro).

Houdini will automatically limit the number of threads to the number of logical processors of your hardware. If your computer supports hyper-threading it is recommended not using more threads than physical cores, as the extra hyper-threads would usually degrade the performance of the engine. See the FAQ about hyper-threading.

The FAQ about computer responsiveness contains some tips about using the correct number of threads to improve the responsiveness of your computer while the engine is running.

Split Depth

When using multiple threads, the Split Depth parameter defines the minimum depth at which work will be split between cores.
Default 10, min 8, max 99.

This parameter can impact the speed of the engine (nodes per second) and can be fine-tuned to get the best performance out of your hardware. The default value 10 is tuned for Intel quad-core i5/i7 systems, but on other systems it may be advantageous to increase this to 12 or 14.

To see which Split Depth value is best for your system you can use Houdini's `autotune` command. Run Houdini in a command window (simply by double-clicking on the executable) and type `autotune` followed by Enter. This command will analyze 4 positions for 30 seconds and measure the average nodes/second for Split Depth values between 10 and 18. Use the value that produces the highest average node speed in your Houdini configuration.
You can set the hash size and number of threads before running the `autotune` command:
- to set the hash size to 512 MB, type `setoption name hash value 512` (alternatively in Houdini 3, type `hash=512`)
- to set the number of threads to 3, type `setoption name threads value 3` (alternatively in Houdini 3, type `threads=3`)

Here's an example `autotune` result on a Core i5-750 using 3 threads and 512 MB of hash memory:

```
Summary:
==========================================================
Split Depth 10, Position 1: 5843 kN/s, idle 8M/s
Split Depth 10, Position 2: 5834 kN/s, idle 10M/s
Split Depth 10, Position 3: 5832 kN/s, idle 9M/s
Split Depth 10, Position 4: 5964 kN/s, idle 10M/s
Split Depth 10 - Average speed: 5868 kN/s, idle 9M/s
==========================================================
Split Depth 12, Position 1: 5849 kN/s, idle 11M/s
Split Depth 12, Position 2: 5742 kN/s, idle 10M/s
Split Depth 12, Position 3: 5712 kN/s, idle 10M/s
Split Depth 12, Position 4: 5881 kN/s, idle 17M/s
Split Depth 12 - Average speed: 5796 kN/s, idle 12M/s
==========================================================
Split Depth 14, Position 1: 5855 kN/s, idle 15M/s
Split Depth 14, Position 2: 5783 kN/s, idle 13M/s
Split Depth 14, Position 3: 5669 kN/s, idle 20M/s
Split Depth 14, Position 4: 5689 kN/s, idle 19M/s
Split Depth 14 - Average speed: 5749 kN/s, idle 17M/s
==========================================================
Split Depth 16, Position 1: 5744 kN/s, idle 25M/s
Split Depth 16, Position 2: 5803 kN/s, idle 19M/s
Split Depth 16, Position 3: 5596 kN/s, idle 34M/s
Split Depth 16, Position 4: 5680 kN/s, idle 33M/s
Split Depth 16 - Average speed: 5705 kN/s, idle 28M/s
==========================================================
Split Depth 18, Position 1: 5721 kN/s, idle 44M/s
Split Depth 18, Position 2: 5781 kN/s, idle 29M/s
Split Depth 18, Position 3: 5521 kN/s, idle 50M/s
Split Depth 18, Position 4: 5546 kN/s, idle 59M/s
Split Depth 18 - Average speed: 5642 kN/s, idle 45M/s
==========================================================
Average CPU time per thread: 597.125 s
CPU-corrected average speed: 5780 kN/s
==========================================================
```

The highest average node speed of 5868 kN/s is obtained with the default Split Depth 10.

3.5 Game Play

Tactical Mode (checkbox)

The Tactical Mode was introduced in Houdini 3.
When Tactical Mode is active, Houdini modifies its search process and uses some clever tricks to focus on tactical solutions in the root position. In Tactical Mode Houdini becomes a very skilled tactical problem solver. Below some results for two popular tactical test suites with Houdini running at 30 seconds per position using 2 threads on a Core i5-750 with 512 MB hash.

1) Arasan 14 Tactical Suite, comprising 197 medium to hard positions
- Houdini 1.5a solves 148 positions, total solution time 34 minutes.
- Houdini 2 solves 155 positions, total solution time 35 minutes.
- Houdini 3 Tactical solves 177 positions, total solution time 22 minutes.

2) Die Tactical Suite, comprising 100 hard positions
- Houdini 1.5a solves 29 positions, total solution time 40 minutes.

- Houdini 2 solves 33 positions, total solution time 39 minutes.
- Houdini 3 Tactical solves 56 positions, total solution time 30 minutes.

Note that the focus on tactical moves will reduce the playing strength of the engine, the Tactical Mode is probably about 30 Elo weaker than the normal playing mode.

One can compare the Tactical Mode to what a human player does when he/she tries to solve a test position in a chess magazine or website. He/she will scan for tactical, forcing moves, threats, sacrifices etc. That way a lot more tactical solutions can be found than what the player would actually see and decide to play over the board, during a normal game. The *"this is a test position"* message changes the search strategy.
Houdini's Tactical Mode works exactly the same, very human-like: it will spend a lot more time looking for tactical moves, threats etc. In most positions this doesn't provide any benefit - most of the time there is no tactical solution available and all the focus on tactics is just wasted. But in test positions (selected because there IS a tactical solution) it works extremely well.

Strength

Limit strength from 0 (beginner) to 100 (full strength).
Default is 100 (full strength).

The strength reduction is mostly based on a combination of two techniques:
- Limiting the number of positions searched - this reduces mostly the tactical strength of the engine;
- Purposely picking a move the engine knows is not optimal - this reduces mostly the positional strength of the engine.

The combination of the two produces a game with both tactical and strategic (positional) flaws.

The following table shows the estimated Elo of Houdini 2 at reduced Strength levels. It is based on the feed-back of several users:

```
Strength  0 => 1200 Elo
Strength 10 => 1500 Elo
Strength 20 => 1800 Elo
Strength 30 => 2000 Elo
Strength 50 => 2300 Elo
```

This table is very approximate and hasn't been validated for Houdini 3 yet.
What is the highest strength level that you can consistently beat?

UCI_Elo

UCI-protocol compliant version of `Strength` parameter.
Default 3000, min 1200, max 3000.

Internally the `UCI_Elo` value will be converted to a `Strength` value according to the table given above.
The `UCI_Elo` feature is controlled by the chess GUI, and usually doesn't appear in the configuration window.

UCI_LimitStrength (checkbox)

Activate the strength limit specified in the UCI_Elo parameter.
This feature is controlled by the chess GUI, and usually doesn't appear in the configuration window.

MultiPV

Number of principal variations shown.
Default 1, min 1, max 32.
The MultiPV feature is controlled by the chess GUI, and usually doesn't appear in the configuration window.

MultiPV_cp

Limit the multi-PV analysis to moves within a range of the best move.
Default 0, min 0, max 999.
Values are in centipawn. Because of contempt and evaluation corrections in different stages of the game, this value is only approximate.
A value of 0 means that this parameter will not be taken into account.

Ponder (checkbox)

Have the engine think during its opponent's time.
The Ponder feature (sometimes called "Permanent Brain") is controlled by the chess GUI, and usually doesn't appear in the configuration window.

Contempt

Level of contempt to avoid draws in game play.
Default 1, min 0 (none), max 2 (aggressive).

The notion of "contempt" implies that Houdini will try to avoid draws by evaluating its own position slightly too optimistically. The Contempt level can be chosen between 0 (none) and 2 (aggressive), the default value of 1 should be a good compromise in most situations.

- 0 = No Contempt
  The evaluations are accurate and identical for both sides. This is recommended for position analysis in which you analyze alternatively for White and Black. The starting position evaluates as approx. +0.15.

- 1 = Default Contempt
  Contempt 1 is primarily based on piece value imbalance, Houdini will value its own pieces higher than the opponent pieces, so will only exchange them if there’s a clear positional advantage in doing so.
  This also means that the score is evaluated optimistically for the side to move (at most 0.15 pawn). For example, the starting position evaluates as approx. +0.30 when analyzing for White and +0.00 when viewed from Black. This is only recommended for position analysis if you always analyze for the same side.

- 2 = Aggressive
  Contempt 2 adds some king safety imbalance, leading to a more attacking style.

The contempt settings are fairly mild and have little impact on the objective strength of the engine. It’s hard to say which will give the best results against a given opponent, it may depend on the style and strength of the opponent. One could envisage more pronounced contempt but this would start to degrade the engine’s objective strength.

By default the contempt is only activated during game play, not during infinite analysis. If you enable the Analysis Contempt checkbox, Houdini will also take into account the contempt for infinite analysis.

Analysis Contempt (checkbox)

Activate Contempt for position analysis.

It is usually not recommended to activate the contempt for analyzing positions.
When contempt is active, the score of the analysis will be optimistic (over-evaluated) for the side that is to move. That means that if you use Analysis Contempt the evaluations will change depending on whether White or Black has the move. For example, from the start position, when you do an analysis with Analysis Contempt (and Contempt value 1) you could find a best move e2-e4 scoring about +0.3 for White. If you then play e2-e4 and analyze for Black you could find a score close to +0.0. If you do the same without Analysis Contempt, you should find a consistent +0.15 score whether it’s White or Black to move.

FiftyMoveDistance

The number of moves after which the 50-move rule will kick in.
Default 50, min 5, max 50.

This setting defines the number of moves after which the 50-move rule will kick in - the default value is 50, i.e. the official 50-moves rule.
Setting this option in the range of 10 to 15 moves can be useful to analyse more correctly blockade or fortress positions:
- Closed positions in which no progress can be made without some sort of sacrifice (blockade);
- End games with a material advantage that is insufficient for winning (fortress).

By setting FiftyMoveDistance to 15, you're telling the engine that if it cannot make any progress in the next 15 moves, the game is a draw. It's a reasonably generic way to decide whether a material advantage can be converted or not.

Mate Search

Maximum search depth for mate search.
Default 0, min 0, max 99.
If set, this option will usually speed-up a mate search.
If you know that a position is "mate in X", you can use X or a value slightly larger than X in the **Mate Search** option. This will prevent Houdini from going too deep in variations that don't lead to mate in the required number of moves.

UCI_Chess960 (checkbox)

Activate Fischer Random Chess a.k.a. Chess960 games.
The Chess960 feature is controlled by the chess GUI, and usually doesn't appear in the configuration window.

3.6 End Game Table Bases

For the **Gaviota EGTB support** you'll need to install the Gaviota EGTB files. At the time of writing they are available for download at the address http://www.olympuschess.com/egtb/gaviota. Download all 145 files and save them in a directory on your hard disk. The total disk space required is about 7 GB.
Alternatively you could skip the download and generate the tables directly on your own computer, please follow the instructions found at the Gaviota web site.

For the **Nalimov EGTB support** you'll need to install the Nalimov EGTB files. At the time of writing they are available for download at the address ftp://ftp.cis.uab.edu/pub/hyatt/TB or http://tablebase.sesse.net. Download all files and save them in a directory on your hard disk. The total disk space required for the 3-4-5-men table bases is about 7 GB (290 files in total).
Alternatively you can order the tables on DVD from several suppliers, this is probably the more efficient way of obtaining the 6-men Nalimov table bases.

For the **Scorpio bitbase support** (introduced in Houdini 3) you'll need to install the Scorpio bitbases. Download the egbb archives with 180 files (230 MB) in 2 parts from http://www.mediafire.com/?du9q8x6nojd3qh6 and http://www.mediafire.com/?5vh54gz9d9cfmh5 and extract the bitbase files on your hard disk. The bitbases are loaded in memory at startup of the engine.

GaviotaTbPath

Folder containing the Gaviota EGTB files.

GaviotaTbCache

Amount of Gaviota EGTB cache memory in MB.
Default 64, min 4, max 1024.

NalimovPath

Folder containing the Nalimov EGTB files.

NalimovCache

Amount of Nalimov EGTB cache memory in MB.
Default 32, min 4, max 1024.

ScorpioPath

Folder containing the Scorpio bitbase files.

ScorpioCache

Amount of Scorpio cache memory in MB.
Default 32, min 4, max 1024.

Hard Probe Depth

Minimum search depth for EGTB probing that can produce a relatively slow disk access, i.e. a "hard" EGTB probe.
Default 24, min 2, max 99.
A value of 24 corresponds to 12 ply deep, meaning Houdini will only probe if the remaining search depth is at least 12 plies.
You can reduce this value if the table base files are installed on a very fast disk, or if you are running less than 4 threads.
For more detailed explanations, see the EGTB support topic.

Soft Probe Depth (for Gaviota or Scorpio)

Minimum search depth for EGTB probing that uses only the memory cache, i.e. a "soft" EGTB probe.
Default 16, min 2, max 99.
A value of 16 corresponds to 8 ply deep, meaning Houdini will only probe if the remaining search depth is at least 8 plies.
You can reduce this value if you are running less than 4 threads. With a single thread a Soft Probe Depth of 6 can be used successfully, with 4 to 6 threads the default value of 16 is more appropriate.
This feature is not available for the Nalimov EGTB, the Nalimov probing code does not support soft probing.
For more detailed explanations, see the EGTB support topic.

3.7 NUMA support (Houdini Pro)

Most CPU mother boards with multiple sockets employ the so-called "NUMA" architecture.
Houdini Pro detects the NUMA configuration at start-up and will adapt its memory management and thread interaction based on the different NUMA nodes that are available.

Numa (checkbox)

Enable or disable the NUMA-awareness. (Enabled by default)
When NUMA is enabled, Houdini will organize memory and threads to take into account the NUMA configuration of the hardware. This can significantly enhance the scaling of the engine beyond 6 cores.
Combined with Large Pages the speed gain can be up to 25% depending on the number of cores, the motherboard and CPU brand.
See the Houdini Pro topic for some real performance data obtained on a 16-core dual AMD Opteron-6128 box.

Numa Offset

Default 0, min 0, max 16.

The NUMA offset is useful if you're running multiple instances of Houdini, in which case you should assign a different NUMA node for each running Houdini process to avoid the Houdini instances competing for the same resources.
For example, on a 16-core hardware with 4 NUMA nodes, you could be running four Houdini processes each using 4 cores. By setting Numa Offset to 0, 1, 2 and 3, each Houdini instance will be running on its own NUMA node without conflicting with the other Houdini instances.

Part IV

4 End Game Table Base support

Houdini integrates the Gaviota EGTB probing code © Miguel A. Ballicora, the Nalimov EGTB probing code © Eugene Nalimov and the Scorpio bitbase probing code © Daniel Shawul.

Gaviota EGTB

For the Gaviota EGTB support you'll need to download the Gaviota EGTB files. At the time of writing they are available for download at the address http://www.olympuschess.com/egtb/gaviota. Download all 145 files and save them in a directory on your hard disk. The total disk space required is about 7 GB. Alternatively you could skip the download and generate the tables directly on your own computer, please follow the instructions found at the Gaviota web site.
The location of the Gaviota EGTB files is specified in the GaviotaTbPath option. Houdini will scan the folder for the "kqkr.gtb.cpX" file and accordingly decide which compression level to use.

Nalimov EGTB

For the Nalimov EGTB support you'll need to install the Nalimov EGTB files. At the time of writing they are available for download at the address ftp://ftp.cis.uab.edu/pub/hyatt/TB or http://tablebase.sesse.net. Download the files and save them in a directory on your hard disk. The total disk space required for the 3-4-5-men table bases is about 7 GB (290 files in total). Alternatively you can order the Nalimov tables on DVD from several suppliers, this is probably the more efficient way of obtaining the 6-men Nalimov table bases.
The location of the Nalimov EGTB files is specified in the NalimovPath option.

Scorpio Bitbases

For the **Scorpio bitbase support** (introduced in Houdini 3) you'll need the Scorpio bitbases. Download the egbb archives with 180 files (230 MB) in 2 parts from http://www.mediafire.com/?du9q8x6nojd3qh6 and http://www.mediafire.com/?5vh54gz9d9cfmh5. The bitbases are loaded in memory at startup of the engine, make sure the physical memory of the computer can accommodate the 300 MB extra.
The location of the Scorpio bitbases is specified in the ScorpioPath option.

Probing Frequency

The frequency of the EGTB probing can be configured with the Hard Probe Depth and Soft Probe Depth options. A "soft probe" will only use the data available in memory, whereas a "hard probe" can require a relatively slow disk access. Soft probing are available only with the Gaviota EGTB.

Houdini will always probe the EGTB for the initial position of the search. If an EGTB mate position is identified, the full main line will be shown immediately. If the initial position is an EGTB draw but Houdini has a material advantage, and if Contempt is active, Houdini will still play for a win - the opponent might be human or an engine without end game table bases.

Question: Should I declare only one type of bases or all three (Gaviota, Nalimov, Scorpio) in the UCI-options?

You should only use one type of EGTB - if you specify several in the UCI options, the last option will prevail.
Houdini cannot use more than one type of EGTB at the time.

Question: Which are the best endgame databases in your opinion?

In our experience all three systems work very well, we don't have any preference. Of course only Nalimov has 6-man bases, but if you're fine with the 5-man tables they're all very good.
Scorpio bases are fine for analysis, but for playing games you may run into some difficulties in the final conversion phase. Inasmuch as the bitbases only contain win/ draw/loss information, it can happen that a won position cannot be converted because it is unclear which move brings the position closer to mate.

Question: I don't understand the concept of setting a depth for EGTB probing. Why wouldn't all positions that have 3-4-5 pieces probe the table bases to get the evaluation?

EGTB probes are extremely slow compared to a normal evaluation by the engine. Suppose you have a K+Q+P against K+N ending. Even without consulting the table bases Houdini knows that this ending is easily won for the K+Q+P side. Consulting the EGTB for this position would reduce Houdini's playing strength, as it could easily have evaluated 1,000 other positions instead of making the rather useless EGTB probe.

Even the in-memory "soft" Gaviota or Scorpio probes are sometimes relatively slow compared to a native evaluation of the position - especially when multiple threads are running. If two threads simultaneously perform a soft probe, the second thread has to wait for the first to finish its probe.

This is aggravated by the fact that Houdini contains a lot of end game knowledge in its native evaluation function - a lot more than most other chess engines. For example, even in a non-obvious ending like KBP v KB with bishops of the same color, Houdini knows quite well which positions are won and which are drawn. This means that except for difficult-to-evaluate endings like KQP v KQ the 5-men EGTB do not really increase Houdini's playing strength.

To handle this situation in an intelligent way, Houdini will vary the frequency at which the EGTB are probed depending on the actual end game that is occurring. In difficult-to-evaluate endings Houdini will probe the EGTB files much earlier than in endings that are easy to evaluate.

The frequency of the probing is further influenced by the following parameters:
- Hard Probe Depth is the earliest search depth at which the EGTB probing will result in a disk access. You can reduce this value if the table base files are installed on a very fast disk, or if you are running less than 4 threads.
- Soft Probe Depth is the earliest search depth for EGTB probing that uses only the memory cache. You can reduce this value if you are running less than 4 threads.

Part V

5 Houdini Pro version

Houdini Pro is intended for power users with high-end hardware.
The main differences with the Standard version are:

- Houdini Pro supports up to 32 threads.
- Houdini Pro supports up to 256 GB of hash memory (262144 MB).
- Houdini Pro supports Large Memory Pages.
- Houdini Pro is NUMA-aware.

Large Memory Pages

Houdini Pro will use so-called large memory pages if they are provided by the operating system. Depending on the hash table size the speed gain may be between 5% and 15%.

To enable this feature in Windows, you need to modify the Group Policy for your account:

1. Run: `gpedit.msc` (or search for "Group Policy").
2. Under "Computer Configuration", "Windows Settings", "Security Settings", "Local Policies" click on "User Rights Assignment".
3. In the right pane double-click the option "Lock Pages in Memory".
4. Click on "Add User or Group" and add your account or "Everyone".
5. You may have to logoff or reboot for the change to take effect.

You'll also need to run your chess GUI with administrative rights ("Run as Administrator") or disable UAC in Windows.
Very often large pages will only be available shortly after booting Windows. After a while the Windows memory becomes too fragmented for large page allocation, and Houdini will fall back to standard memory page usage.

You can test the availability of Large Pages with the `lp` command. Run Houdini in a command window (simply by double-clicking on the executable) and type `lp` followed by Enter. Houdini will produce a summary with the number of allocated large pages as a function of the large page size. This command can take several minutes on a system with lots of ram (16 GB or more), so be patient.

NUMA-awareness

Most CPU mother boards with multiple sockets employ the so-called "NUMA" architecture.
Houdini Pro detects the NUMA configuration at start-up and will adapt its memory management and thread interaction based on the different NUMA nodes that are available.
Speed gain can be 5% to 15% depending on the number of cores, the motherboard and CPU brand.

Running Multiple Houdini Pro instances

If you're simultaneously running multiple Houdini Pro instances they will by default compete for the resources on the same NUMA nodes. To avoid this, you should set the Numa Offset parameter to different values in the different Houdini instances.

For example, if you want to run two Houdini instances with 6 threads each on 12-core hardware, you should use Numa Offset 1 for the second instance so that it will allocate its 6 threads on the second NUMA node. See also the Numa Offset configuration.

Some Real Performance Data

The test system was a 16-core dual AMD Opteron-6128 box running at the stock 2.0 GHz speed.
The `autotune` command (see the topic on Split Depth) was used as benchmark to measure the impact of the Large Pages and the NUMA-awareness.
Hash memory was set at 2048 MB, 16 threads were used.

Configuration	Best Split Depth	Average Node Speed	Speed Gain
Standard	14	13600 kN/s	
With Large Pages	14	14900 kN/s	+10%
With NUMA and Large Pages	12	16200 kN/s	+20%

On this system Houdini Pro with NUMA and Large Pages was about 20% faster than the Standard version.