

The Hive Workshop Wish-list

The Hive Workshop (or THW) is a Warcraft 3 and Starcraft 2 modding community founded in 2004. We have over 92,000 members and a vast resource section, and we're currently the largest and most active Warcraft 3 modding community. We are all really excited for what may await Warcraft 3, and we're more than willing to provide suggestions and feedback towards resurrecting the game.

The document is split into four categories:

1. Battle.net suggestions - hosting and exploiting
2. General Suggestions - ideas that would bring players back to the game
3. Extensive Tasks - tasks that are important but may be a lot of work
4. Map-making and Miscellaneous - suggestions to give map-makers more powers

On page 9, there is a summary of battle.net's problems, mostly in regard to host bots. Page 10 has useful references.

Thanks again for taking the time to consider our list!

Sincerely,
The Hive Workshop Team

Low Priority

High Priority



Link



Battle.net Suggestions

A summary of the current problems related to battle.net are listed on page 10.

1. **Improve Custom-Game Hosting:** hosting is too difficult, so people have been using host bots. This leads to empty lobbies and an environment that does not cater to new/returning players.
 - Port forwarding and firewall issues are major drawbacks to the default wc3 hosting system.
 - Hosted games should be visible on all realms (bots already provide this)
 - Incorporating battle.tags would be a huge plus
2. **Update Warden/Active Support For Regulating Hackers:** if melee players are to go back to the original ladder, then there has to be a way to deal with map-hackers. Player-based reporting (with game master's to handle those cases) will also go a very long way, as illustrated in the existing communities.
3. **Generally Reduced Delays:** the way to approach this is not clear, but many of the complains regarding battle.net have to do with the delays. One thing that would help is to host games on a server rather than synchronizing them through the 'host' user. This may also help with reconnection problems.

General Suggestions: Bringing Players Back

1. **New Content:** Starcraft 2 recently introduced the idea of “Mission Packs”. We feel that a similar concept could get players interested in Warcraft 3 again. A new bonus campaign or even just a new tavern hero would have people coming back to reinstall. It could even just be a map with Chris Metzen speaking in Thrall’s voice for 4 hours, we’re positive that people would buy it.
2. **Add Warcraft 3 to the Battle.net App:** ideally, patching/installation/installing maps would all be simplified. Incorporating battle-tags would do wonders as well. It could appear in a “classics” game tab if necessary.
3. **Widescreen Support:** this suggestion comes up a lot. Warcraft 3 supports higher resolutions as of patch 1.25b, but it does not have the correct aspect ratio—it just stretches the display.
4. **Community Contests and Support:** who doesn’t miss the old classic battle net? I’m sure players and modders alike would love to see new tournaments and community map contests. We definitely have plenty contests of our own, but the Blizzard-hosted ones get tons of hype.
5. **Remastered Cutscenes:** anyone who has played Warcraft 3 will remember its cutscenes. At the time, the quality blew everyone’s minds. Sadly, Warcraft 3 has them highly compressed so it could stream it without buffering. It would be great to see them in HD or a remastered form.

Extensive Tasks

1. **Modern Operating Systems:** Warcraft 3 has some bugs with the latest Windows (mostly related to newer Direct3D systems). Some players have explicitly put d3d8.dll or d3d9.dll (old DirectX versions) in their folder to fix some of the issues. Warcraft 3 also has a slew of issues on OS X; to name a few: the installer and editor is still PowerPC (cannot be run on OS X 10.7+), battle.net and LAN do not work on OS X 10.9+, among other bugs. In addition, the CGI cinematics have had a lot of issues loading on both operating systems (e.g. black screen or crashes). Relevant pages are linked towards the end of the document.
2. **Battle.net 2.0:** it is a bit of a buzz word, but a lot of players hoped that Warcraft 3 would get a networking overhaul since delays are so high. Battle-tags would also be a fantastic addition.
3. **Allow For Dynamic UI Management:** both in-game and just in general. At the moment, the game just renders black-bars behind the UI, so if you try to fiddle with the UI, you'll still be stuck with those bars. Ideally, the game would support better handling of *.fdf files, perhaps even within game (similar to Starcraft 2 'dialogs' or WoW's UI). Other possibilities would be to dynamically set tooltips in-game (they are currently cached/fixed within the war3map.wts for a given map), edit the mini-map as well as the individual unit textures, and options for fiddling with the command-button panel or the unit pane/portrait.
4. **General Engine Improvements:** the engine for Warcraft 3 was made at a different time for a different time. In general, the engine doesn't handle too many objects on the screen all that well, lighting can appear blocky omnilights intersect, high-polygon models can experience really weird graphic bugs (e.g. vertices stretching to infinity), and destructables are rendered even when they are not on-screen (this might have a particular reason, though).
5. **Bump Map/Normal Map Support:** in order to support this seamlessly, there could be an 'expected path' where the texture would be stored. For example, icons are expected to be placed under 'ReplaceableTextures\CommandButtons\BTNxxx.blp' so that it can look for additional passive/disabled textures associated with it. A similar system could be made for bump-maps and normal-maps so that they could be applied to existing models easily.
6. **Load a Map For Multiple Players:** multiplayer campaigns. It would be cool to transfer players from one map to the next, similar to how it is done in Rexxar's campaign, but for all players. At the moment, everyone has to copy a save code, leave the game, make a new game, join, and type up the code in order to play multiplayer campaigns.

Bugs and Exploits

1. **Clear Warcraft 3's Cache between Maps:** if you try to play map X and then play map Y (in the same game session), there is a chance you'll instantly disconnect from the game due to memory left-overs from map X. It forces a player to restart just to play map Y. This mostly happens with custom games and it makes arranging games a mess (players will instantly leave).
2. **"Preload Exploit":** the preload exploit abuses Warcraft 3's Preload() function to execute external scripts or write text files. Most people have used it to store save-load codes for multiplayer maps, but it is possible to abuse it. It is on a similar caliber to the "return bug" exploit. Instead of removing the preload bug (since some games already use it), we suggest that you limit the file extension to *.txt or *.pld and limit the creation to being within Warcraft 3's folder. Alternatively, getting a 'bank' system (like Sc2) to easily store/read data on disk in multiplayer would be a fantastic substitute.
3. **"Saved Game" Bugs:** when loading a saved game file, a couple of bugs can occur: (1) periodic timers in the map turn into non-periodic timers (2) the abilities within a spell-book reset to level 1 (3) accessing an array index of 8191 before saving will cause the loaded game session to crash (4) lightning handles aren't saved correctly; this is problematic because operating on destroyed lightning causes the game to crash.
4. **Maps with Long Names:** maps with very long names or very long paths relative to the /Maps/ folder will not appear listed when you try to select them from the "Custom Game" menu in-game.
5. **Pan Camera as Necessary:** this trigger-action will instantly cause players to disconnect. Most modders have gotten used to avoiding it, but it is a very easy bug to fix (described [here](#)).



Map-Making and Miscellaneous Suggestions

1. **Avoid Breaking Existing Maps:** when the famous “return bug” was fixed in patch 1.24, tons of maps broke (including DotA). Fortunately, Blizzard gave us new functions to transition our maps, but the change still left its mark. Some of the community became divided because players wouldn’t be able to play some of their favorite maps without down-patching (i.e. if the author was inactive). On a similar note, many map-makers currently use an extended editor “JassNewGenPack” to create their maps. It simply adds editor functionality for things that are already supported by Warcraft 3. If you ever re-compile the editor, we ask that you keep ‘worldedit121.exe’ available in order for people to transition their maps.
2. **Increase the Map Size Limit on Battle.net and LAN:** the current limit is 8 MB. It would be great to have this extended, e.g. 64 MB or 128 MB. This is presumably a simple fix considering the limit was extended from 4 MB to 8 MB in patch 1.24b. This change would definitely excite modders.
3. **Dynamically Change Fields At Runtime:** the following is just a list of commonly requested properties that map-makers would enjoy having access to via *triggers*:
 - Get and set a unit’s ability/item cooldown time
 - Get and set attack damage, armor, attack range, attack speed, attack index, backswing, etc.
 - Get and set object editor fields at runtime, e.g. a unit’s model, a unit’s attack missile, set tooltips for units/abilities/items, set icons, etc.
 - Set a unit’s facing, pitch, and roll instantly (setting a unit’s facing currently has turn-speed delay)
 - Better control over a unit’s portrait
 - Set a unit’s maximum hit points and mana points
 - Get a unit’s damage type
 - ‘Attack released event’ - an event that fires when an attack is released i.e. released projectile
4. **Mouse Tracking:** natives to detect the coordinates of the local player’s mouse on screen and on the map, mouse click/drag/release events, show and hide cursor, IsMouseOverUI(), etc. Synchronization is a problem, but that can be left up to the map-maker.
5. **Keyboard Tracking:** natives to detect when a player presses a key. Currently, Warcraft 3 allows us to detect when arrow-keys are pressed, but they experience big delays due to synchronization. Even in single-player, the key detection *still* gets synchronized. You can leave the current arrow-key implementation as is for backwards-compatibility, but it would be great to be able to detect key press and depress and leave it unsynchronized. Synchronization can be left up to the map-maker.

6. **Native to Remove Trackables:** trackables allow map-makers to create models in-game and detect when a player clicks or hovers over it (e.g. a sign-post). The biggest issue is that they cannot be destroyed through triggers.
7. **Ability Control:** natives/*triggers* to control abilities. For example: setting cooldown, setting mana cost, setting tooltip, setting button position, setting hotkey, hide an ability, etc. It would also be great to be able to show/hide the standard command buttons (stop, move, patrol, etc.).
8. **Allow Users to Change Item Hotkeys:** particularly in CustomKeys.txt. Many players play without numpads, especially laptop users.
9. **Improved Image API:** images exist in Warcraft 3, but they end up stretching edge pixels to a particular size. We end up having to add a transparent border to image files, which works, but it takes up extra space and it means we can't use default in-game icons/images. In addition, it would be fantastic to be able to display images on screen and be able to interact with them, such as with the "idle worker" icon or the hero icons that appear on the upper-left part of the screen.
10. **Unit Expiration Timers:** these cannot be canceled (e.g. UnitApplyTimedLife()).
11. **Increased Movement Speed Limit:** the current limit is 522.
12. **Get Player Resolution:** a native to retrieve a player's Warcraft 3 display resolution. This would mostly be for in-game UI and for multi-board adjustments.
13. **Script-Related:**
 - Allow for arrays to be passed by reference (e.g. as an argument for a function)
 - Allow function pointers of an arbitrary type ('code' only supports 'takes nothing returns nothing')
 - It would be great if function pointers ('code') could be callable directly. e.g. code c -> call c()
14. **Better Graphic Options:** it would be great to have an in-game option to enable anti-aliasing and have the view distance limit (far-Z) extended from 10000 to something higher.

15. **Various Bugs:** these are various bugs of minor importance:

- When a unit is paused, it cannot accept power-ups. It still fires an item-pickup event though. The item drops to its previous location.
- If you try to move or destroy a lightning handle that has already been destroyed, the game will crash. Ideally, it should do nothing or report an error.
- Accessing an invalid player index through the Player() native will crash the game.
- If you use GetItemType(item) on an item of type "Miscellaneous", it will return ITEM_TYPE_UNKNOWN, not ITEM_TYPE_MISCELLANEOUS.
- If you make a local variable in JASS that points to an *agent*, it increases that agent's reference count by 1, but that count never goes down unless you null the local variable. This prevents the agent's ID from being recycled, which is a tiny memory leak. Ideally, local variables would automatically be 'nullified' once execution reaches the end of a function, since that is the scope of a local variable.
- Terrain deformations are never removed. They can be stopped, but they still take up memory. Tomes experience a similar issue where the actual item never ends up removed.
- If a unit has a collision size greater than or equal to 16, its model is shifted slightly.
- ShowInterface(true, 0) mistakenly hides all portrait models.
- "Fogged sky" will be properly covered by terrain fog in the editor, but in-game it is uncovered.



Battle.net Summary

Warcraft III has a bot problem. However, not all of the bots are bad. There are four main categories of bots: host bots, clan/chat bots, spam bots, and various map-hack-related bots.

1. **Host bots:** host bots arose for various reasons:

- (1) Hosting a custom game is complicated by today's standards (port forwarding)
- (2) Bots can host cross-realm (US West, US East, EU, etc.)
- (3) Bots have tons of special commands (banning, kicking, swap player slots, etc.)
- (4) Bots can yield lower latencies (compared to traditional hosting) and help facilitate reconnections when a player temporarily loses connection
- (5) Open games are displayed on a website (e.g. MakeMeHost), easy searching
- (6) Some are used to regulate trolls, hackers, and spammers.

Host bots are a blessing and a curse. They have kept the game alive, but they do not cater to new/returning players. There are often more bots than players and games that stay empty for hours, particularly games that are "auto-hosted" rather than having a dedicated person sitting in the lobby. Ideally, battle.net would be revamped to lessen the need for host bots. But at the end of the day, Blizzard shouldn't completely remove host bots. They'll go away naturally if Blizzard provides better alternatives.

2. **Clan/Chat bots:** these bots are typically used for whispering/general communication. For example, a clan can whisper people joining a channel to join their clan. Some are used to notify newcomers of events or for clan trivia.

3. **Spam bots:** these bots allow users to automatically send a message at a given interval, bypassing battle.net's default spam filter. Many user's ignore lists fill up, and this is another reason why people migrate towards community-regulated networks (e.g. w3arena).

4. **Map-hacks:** Not all of these are bots, but they're two sides of the same coin. Some are used to abuse match-making (Loss-bots), some are used to remove fog, some will force users to experience lag (lag-hack), etc. These players have already filled the ladder, so most competitive players migrated to regulated communities (w3arena, NetEase).

Further Reading

1. [Previous Wish List](#)
2. [More Host-Bot Perspective](#)
3. [Warcraft 3 Crash Logs - Compilation of Custom Game Crash Logs](#)
4. [Why play on W3Arena?](#)
5. [Sample Host Bot Commands](#)
6. [Preload Exploit](#)
7. [OS X Client Issues - Some Windows 10 Issues](#)
8. Throughout this document, the word “native” and “trigger” are mentioned a lot. Triggers are used to “program” functionality into maps, similar to the way Lua is used to create boss fights in WoW. Natives refer to the specific *functions* that are available to us, e.g. KillUnit() to kill a unit. We may use these interchangeably, but we ultimately just want particular functions exposed to us for scripting purposes.
9. [MakeMeHost](#)
10. [EntGaming](#)