

BATTLETECH MOD SUPPORT

Hello, and thank you for playing BATTLETECH! This document is meant to offer a basic primer on how to successfully manage mods within the game. Mod creation is still a fan driven effort, and while we have included some resources to get you started, the game itself provides no built-in resources to generate modded content.

The mods themselves must be compatible with the latest version of BattleTech, and even then there is always a chance that combining mods can create a non-functional game! In general, mod at your own risk. **As mods are created externally to the game, not all mods are guaranteed to work.**

1. Mod Management

The Mod Management screen is accessible from the main menu. When the mod system is enabled, it builds a list of all the installed mods and displays them here. On this screen you can enable/disable the mod system as a whole, or toggle on or off individual mods. Enabling a mod that has a dependency on other mods will automatically enable any dependent mod. And disabling a mod will also automatically disable any mods that depend on it.

1.1 How to install a mod into BattleTech

- 1) Find a mod you're interested in. You can start with some common modding communities, listed below.
<https://www.nexusmods.com/battletech>
<https://www.reddit.com/r/BattleTechMods/>
- 2) Once you've identified and downloaded a mod, place the mod contents into a named folder inside the same folder where this document is located.
 - a) WIN: C:\Users\user_name\MyDocuments\MyGames\BattleTech\mods
 - b) LIN: /home/user_name/MyGames/BattleTech/mods
 - c) OSX: /Users/user_name/MyGames/BattleTech/mods
- 3) Launch the game, then click the 'Mods' button at the bottom left of the main menu
- 4) You should now see your new mod in the list. If so, it's installed and ready to go!

Note that the first time you install mods, it may take some time to load the mod.

1.2 How to uninstall a mod

- 1) Delete the mod you inserted into the BattleTech/mods folder in step 2 of installation
- 2) Launch BattleTech, and the mod should now be removed from the list!

PLEASE NOTE - Mods affect saves. If you save your game with mods installed, those saves will no longer be loadable if you uninstall the required mods.

1.3 How to change the mod installation directory

- 1) The game expects a file named “modFolderRedirect.txt” to be placed in the root of its persistent data folder. Depending on your system, this may be in different places:
 - a) WIN: %userprofile%\AppData\LocalLow\Harebrained Schemes\BATTLETECH
 - b) LIN: \$XDG_CONFIG_HOME/unity3d/Harebrained Schemes/BATTLETECH
 - c) OSX: Unity scans multiple folders. If you find multiple directories, know that Unity favors the oldest timestamp.
 - i) ~/Library/Application Support/Harebrained Schemes/BATTLETECH
 - ii) ~/Library/Application Support/unity.Harebrained Schemes.BATTLETECH
 - iii) ~/Library/Caches (This is unlikely)
- 2) The entire contents of this folder should be an *absolute path* to the folder you want to use; e.g. C:\BTMods. Please note that when redirecting mods, the BattleTech sub folder is not created.
- 3) If the file, and the path it specifies exist and are accessible to the current user, mods will be loaded from this folder.

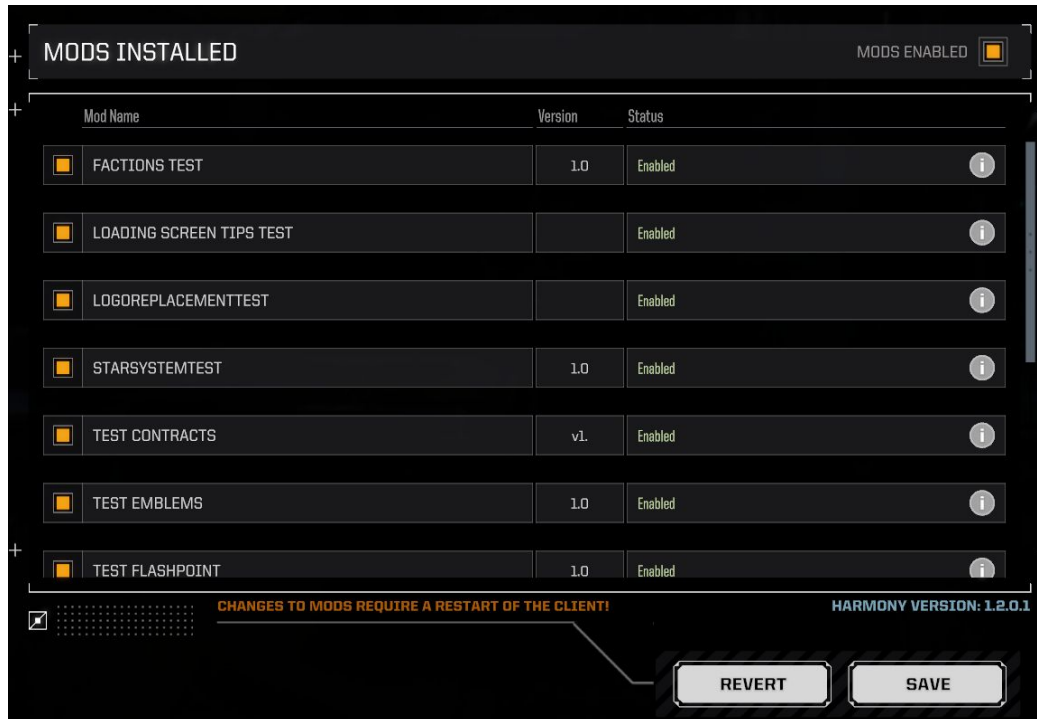
OSX Note: To create a plain text file using TextEdit, use the following steps:

1. Launch Text Edit
2. Create a new file (File > New)
3. Change it to plain text format (Format > Make Plain Text)
4. Set your absolute path as desired in this file
5. Save as modFolderRedirect.txt in the appropriate directory

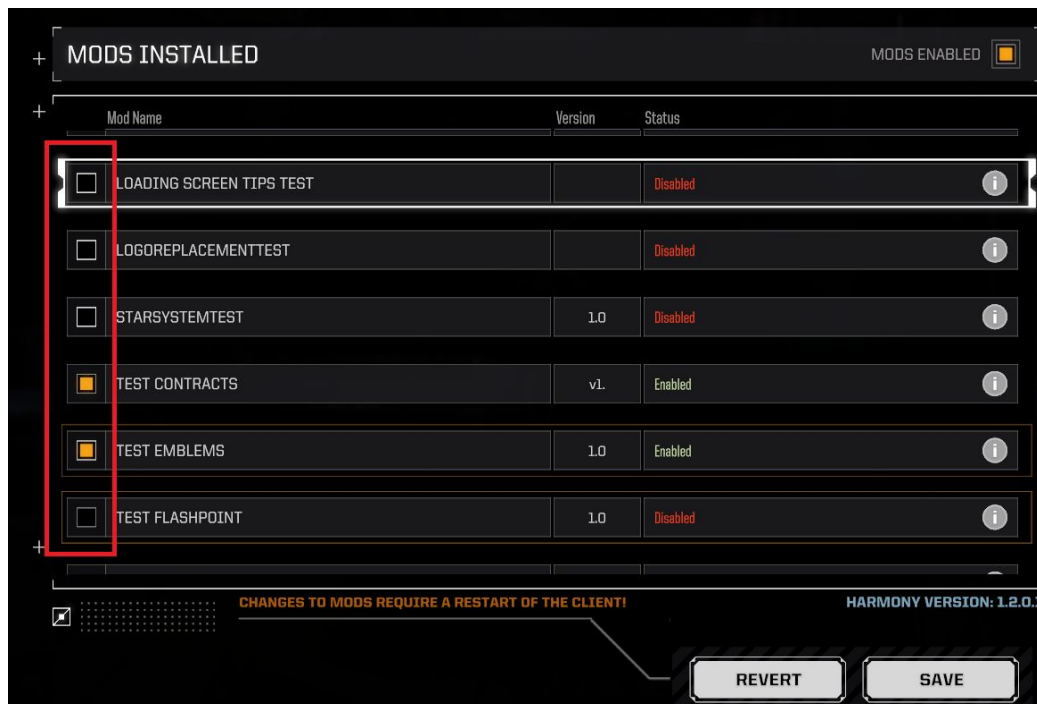
1.4 Enabling and disabling mods

Mods are enabled when they are installed by default. However, they can be disabled after they are installed, in case you want to play without your installed mods active. (See above note in 1.2 about save compatibility)

When mods are active they will be marked “Enabled” in the status bar:

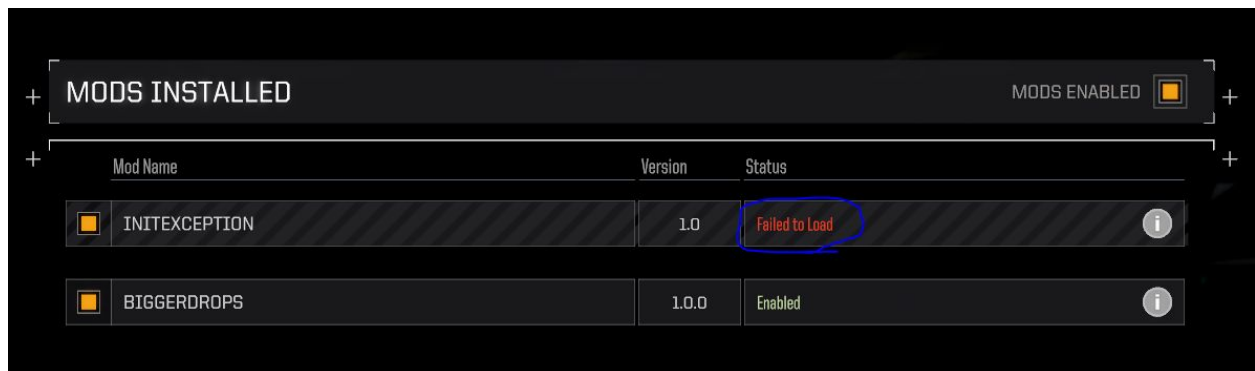


To disable mods, check the box next to the mod, and it will be marked “disabled”:

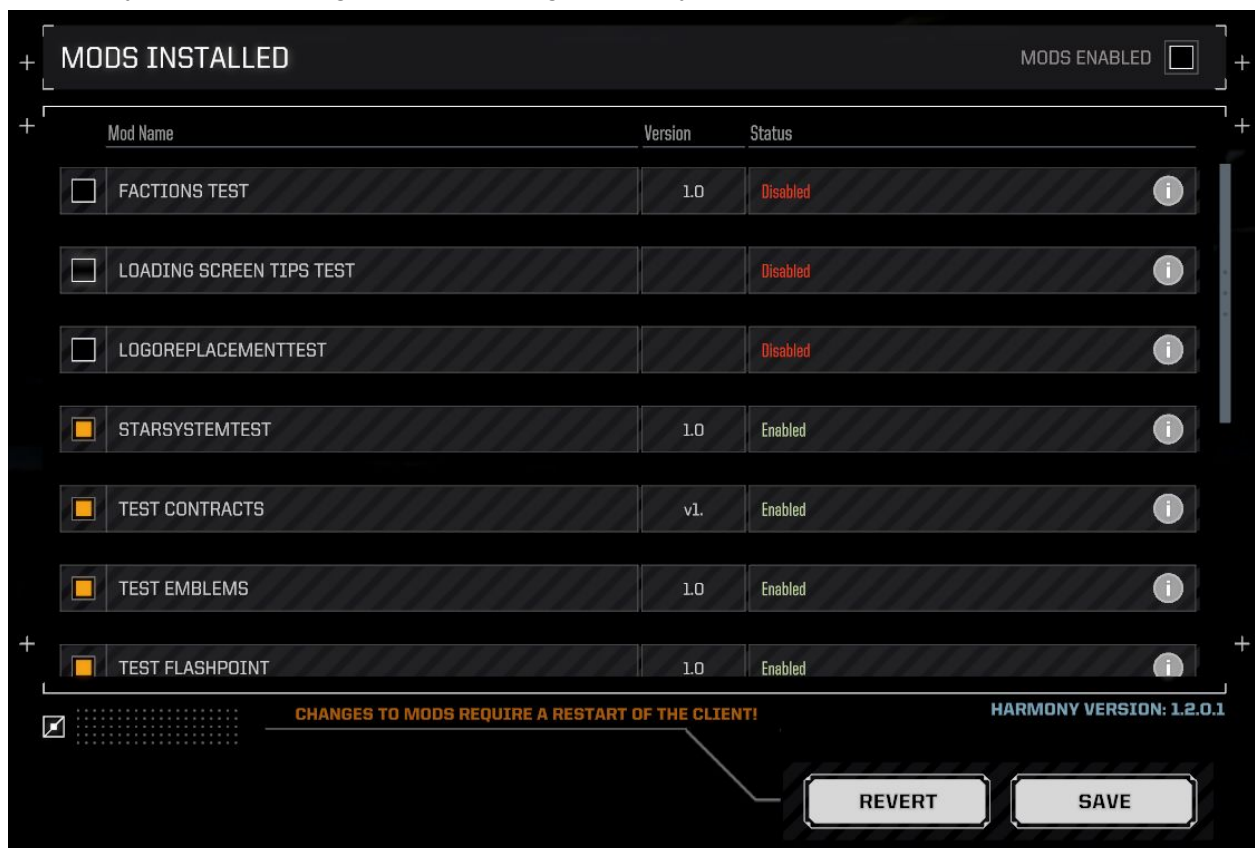


If a mod is enabled but fails to load (such as in cases where the mod version is outdated), it will be grayed out and marked as “mod failed to load”. As long as the mod remains enabled, the

game will try to load the mod on each pass. However, if the mod json is unparsable (such as having bad json), then the mod will not show up in the list at all.



If you want to disable all mods, uncheck the “Mods enabled” in the top right corner. This will put an overlay over all existing mods, disabling the ability to interact with them.



Disabling the mod loader via the “mods enabled” checkbox respects the last known status of the existing mods. This means that the next time you load the game, mods will not be loaded, but their previous status will be retained should you re-enable mods. Changes to

installed/uninstalled mods will not be reflected until the “mods enabled” checkbox is enabled and the game rebooted.

As a reminder, ModLoader will show the last known status of mods, but is not designed to verify whether a mod will work. Mods are external to the loader, so mod at your own risk

You will need to restart the game for changes to take effect!

1.5 Modding Gotchas

- 1) Whenever mods are enabled or disabled, the game must be restarted in order for the changes to take effect. You will be prompted to restart the game if you make changes from the Mod management screen.
- 2) Once the game is modded, any Game Save produced will contain modded data and the mods you have installed are now required to load that save game. The save system is mod aware, and will lock out saves that include modded data, if those mods are not enabled in the mod manager.
- 3) If you alter the contents of the Mods folder on disk while the game is running, you will have to relaunch BattleTech for changes to be detected.
- 4) It is highly recommended that when playing multiplayer and using mods, that both users have the same mods installed. Not having the same mods installed will create a disconnect between game clients and prevent users from launching multiplayer matches.
- 5) It is possible for mods to break a BattleTech install. If you believe your game is in an unrecoverable state, do a clean reinstall of BattleTech to restore the game back to it's previously unmodded state.

2. Getting Started with Modding

2.1 Basic Concepts

Mods are made up of a **JSON** file that describes them, and a collection of assets. Sometimes those assets are a **DLL** that is patched into the game at runtime, and sometimes they are actual game assets that overwrite existing assets, or add new ones. This document primarily describes the **Mod JSON Format**, and the difference between **System Mods** and **Game Mods**.

2.2. Mod Types

Mods come in two flavors. There are **System Mods** and **Game Mods**. The only way that the ModLoader knows the difference between the two is from the name of the **.json** file used to load

them. Without a “**systemMod.json**” or a “**mod.json**” the game will have nothing to point it towards the files it should load.

Mod Type	Mod Filename
System Mod	systemMod.json
Game Mod	mod.json

2.3 Mod JSON Format

```
{
  "Name": "MyLoggerSystemMod",
  "Enabled": true,

  "Version": "0.1.01a",
  "Description": "A system mod to add new logs to ModLoader",
  "Author": "",
  "Website": "https://wewantedmoreloggers.com/morelogs",
  "Contact": "loggermail@wewantedmoreloggers.com",

  "DLL": "MyLoggerSystemMod.dll",
  "DLEntryPoint": "MyLoggerNamespace.MyFlanelClass.MyAxeMethod",

  //Game Mods Only
  "Manifest": [ { "Type": "MechDef", "Path": "NewMechDefPath" }, ],
}
```

2.3.1 Information

These are optional fields that are used to fill out some of the information located in the **Mod Manager Screen**. The rest is potentially good information for mod users if you are looking to have feedback directed somewhere.

Field	Type	Description
Version	string	Version of this mod. Displayed in the “Version” column of the Mod Manager Screen
Description	string	A short description of this mod

Author	string	Author of this mod
Website	string	Website or post that users can go to for more information. Clicking on the Info Icon of the Mod Manager Screen will attempt to take you there
Contact	string	Contact info for the mod author
PackagedOn	DateTime	The date/time that this mod was packed on

Information Sample

```
{
  "Name": "MyLoggerSystemMod",
  "Enabled": true,
  "Version": "0.1.0",
  "Description": "A system mod to add new logs to ModLoader",
  "Author": "",
  "Website": "wewantedmoreloggers.com/morelogs",
  "Contact": "loggermail@wewantedmoreloggers.com",
  "PackagedOn": "2020-02-02T00:22:02Z"
}
```

2.3.2 Enabled

The enabled field will tell the **ModLoader** to either load your mod or to ignore it. If the Enabled field is missing, the ModLoader defaults to trying to load the mod. Whether or not the field has been set, the user will be able to toggle a **Mod** on and off from the **Mod Manager Screen**. Setting whether or not a mod is enabled through the Mod Manager Screen **does not** write the new value to the **Mod Json**. Instead, this is stored in the game data as part of the **Mod Status**. If the Mod Json does not match the stored data, the file setting wins.

Please note that if other mods are dependent on a mod, they will also be disabled if their dependency is disabled or fails to load.

Enabled Sample:

```
{
  "Name": "MyAwesomeMod",
  "Enabled": true
}
```

```
}
```

2.3.3 Dependencies

These fields determine if **Modloader** should load your mod based on the other installed mods, and what order they should be loaded in. It should be noted that **Game Mods** and **System Mods** can depend on other System Mods. However System Mods cannot be dependent on Game Mods. This is due to System Mods completing their load process before Game Mods are loaded.

Dependencies will also be indicated in **Mod Manager Screen** when hovering over the info icon. This can be used as an indicator for why a mod may or may not have loaded depending on what you have installed or enabled.

Field	Type	Description
DependsOn	array <string>	Don't load this mod if these mods are not loaded and load my mod after these mods. Hovering over the Info Icon in the Mod Manager Screen will display these
OptionallyDependsOn	array <string>	Load this mod <i>*after*</i> these mods if they're in the load order
ConflictsWith	array <string>	Don't load this mod if these mods are in the load order

Dependency Sample

```
{  
    "Name": "MyLoggerSystemMod",  
    "Enabled": true,  
  
    "DependsOn": [ "Dependency", "Dependency2" ],  
    "ConflictsWith": [ "PotentialConflict" ],  
    "OptionallyDependsOn": [ "OptionalDependency", "OptionalDependency2" ]  
}
```

With dependencies, you can create a situation where Mod A depends on B, B depends on C, and C depends on A. When we detect a circular dependency, none of the mods can load. If

there's a circular **Optional** dependency, it loads them in an unspecified order. The modloader.log file should report any problems that it finds. There may exist some strange combination of optional dependencies vs regular dependencies that confuse the ModLoader. Use dependencies responsibly.

2.3.4 DLL

The DLL field points to the packaged DLL that the mod is intended to load and patch into the game. This is optional in regards to **Game Mods** but required for **System Mods**. Without a DLL a System Mod cannot really do anything.

Field	Type	Description
DLL	string	Path to .dll file that should be loaded relative to the mod's directory
DLL Entry Point	string	The method that you want ModLoader to call when your DLL is loaded. Defaults to calling Init() if not set

DLL Sample

```
{
    "DLL": "MyLoggerSystemMod.dll",
    //Optional
    "DLLEntryPoint": "MyLoggerNamespace.MyFlanelClass.MyAxeMethod",
}
```

3. ModLoader Patching

The ModLoader may be patched using **System Mods**. System Mods are a type of mod that are loaded prior to the standard **Game Mod**. Because of this, they can go so far as to patch how Game Mods will be loaded. System Mods, unlike Game Mods, are exclusively a compiled **DLL** paired with the **SystemMod.json** that points the game to it. A writeup on how to get started working on DLL based mods can be found in the “Additional Resources” section at the end of this document.

3.1 System Mods

Mods can be made up of a **Game Mod**, a **System Mod**, or both! System Mods load and patch before Game Mods have started loading, so they can have an effect on that process. System Mods do not have a **Manifest** as they are intended to be **DLL** only mods and not introduce new assets or content. Otherwise, **SystemMod.json** is formatted and loads very similarly to **mod.json**.

Things that can potentially be done with System Mods:

- Add new logs to the Game Mod load flow
- Manipulate how ModLoader loads Game Mods

4. ModLoader Configuration

When you disable mods, it keeps a cache of what was installed and enabled at that time. Any changes to installed mods won't be reflected until the mods system is reenabled. This is kept in **mod_status.json** and **system_mod_status.json** files in the **mods\HBS\Cache** directory.

There are also a few settings in the **settings.json** file that will be useful for modders. This can be found in your BattleTech installation directory.

For OSX users, navigate to your BattleTech application folder. Then right-click the BattleTech application and select "Show Package Contents." From here you can navigate the rest of the way to the **settings.json**

WIN: **BATTLETECH\BattleTech_Data\StreamingAssets\data\debug\settings.json**

MAC: **BATTLETECH\Contents\Resources\Data\StreamingAssets\debug\settings.json**

```
"modLoggerLogLevel": "Error",  
"alwaysClearModCache": false,  
"disableSplashScreens": false,  
"disableIntroMovie": false,
```

- **modLoggerLevel** - defaults to Error, and determines what types of things are exported to the logs. Valid options are Debug, Log, Warning, and Error.
- **alwaysClearModCache** - defaults to false - The modcache system exists to speed up subsequent playthroughs for modded games. While developing mods, you might find it useful to have the cache always be cleared to prevent stale data.
- **disableSplashScreens**, **disableIntroMovie** - both default to false - to improve iteration times when developing mods, turn these off so you don't have to wait before the mods start loading

4.1 Clearing the mod cache

There's a setting in settings.json `alwaysClearModCache` which is off by default. When developing/debugging mods, it might be useful to turn this back on by setting the value to true.

The ModCache will be cleared if installed mods go missing, new mods get installed, if the version of the game changes, or if the `alwaysClearModCache` setting is enabled.

5. Additional resources

5.1 Popular Mods

Nexusmods is a common resource for Battletech mods.

<https://www.nexusmods.com/battletech?tab=popular+%28all+time%29>

5.2 Mod Communities

There are several communities where you can talk about mods and modding, including recommendations and troubleshooting. The Paradox forums also link to various external communities (such as Discord channels) that you can join for further conversations.

<https://forum.paradoxplaza.com/forum/index.php?forums/battletech-mod-forum.1012/>

5.3 Mod Creation

Want to learn how to make mods? Talk to the communities linked above, and check out these articles on different ways to make mods:

<https://github.com/BattletechModders/ModTek/wiki/The-mod.json-Format>

<https://github.com/BattletechModders/ModTek/wiki/Writing-ModTek-DLL-mods>

<https://github.com/BattletechModders>

<https://github.com/BattletechModders/ModTek>

5.4 DLC Design Data

If you want to see how we do things under the hood, HBS hosts DLC design data here:

<https://github.com/caardappel-hbs/bt-dlc-designdata>