# Developing the Lionheart Love Patch for WoW Classic: A Technical Guide

## I. Introduction to WoW Classic Modding

World of Warcraft Classic, launched in 2019, meticulously recreates the original 2004 massively multiplayer online role-playing game (MMORPG) in its vanilla state, prior to the release of its first expansion, The Burning Crusade. This recreation features a level cap of 60, deliberately excludes content from subsequent expansions, and faithfully replicates most of the original gameplay mechanics, albeit with some modern interface enhancements and bug fixes.[1] The enduring demand for Classic servers within the community arose from a widespread preference for older game versions, as numerous expansions had progressively altered or removed original content and introduced mechanics that many players found controversial.[1] Blizzard has since diversified the Classic experience, offering various server types such as Classic Era, which maintains the 1.12.1 state indefinitely, and time-limited "Season of Mastery" servers, which concluded in February 2023 after introducing more challenging mechanics. Additionally, official Classic Hardcore servers, featuring permadeath, were launched in August 2023.[1] The term "Classic" has also expanded to encompass subsequent expansions released in their original forms, including Burning Crusade Classic, Wrath of the Lich King Classic, and Cataclysm Classic.[2]

Historically, before Blizzard introduced official Classic servers, players often resorted to unofficial emulated servers to experience the original WoW. These servers, while providing a glimpse into the past, were frequently unstable, legally ambiguous, and imperfect recreations of the authentic experience.[3] A significant breakthrough by Blizzard eventually enabled them to run Classic servers on modern infrastructure, though these official Classic realms intentionally omit certain contemporary features, such as cross-realm servers, Looking For Raid (LFR), and Dungeon Finder (DF) automatic party matchmaking.[3] This deliberate absence of automated social and group-finding features in the official game creates a compelling rationale for the development of custom systems, such as the proposed "Lionheart Love Patch," within a private server environment.

The comprehensive vision for the "Lionheart Love Patch," encompassing a matchmaking system, love-themed events, romantic interactions, mini-games, and lore-based quests, inherently requires a level of game modification not supported by official Blizzard servers. The inherent limitations of official WoW Classic, which by design lacks modern matchmaking and cross-realm functionalities, directly necessitates the use of private servers for ambitious custom content. The private server ecosystem provides the essential flexibility and granular control required for deep game modifications, effectively filling a void left by Blizzard's focus on replicating the pure "vanilla" experience. Consequently, the development and deployment of such a comprehensive patch must occur within a private server environment, as official servers do not permit the extensive alterations to game logic, new systems, or custom assets required. This also means that the target audience for this custom patch is exclusively players within the private server community.

### Understanding WoW Classic's Core File Formats

World of Warcraft Classic relies on a diverse array of proprietary file formats to organize and store its vast game data. A foundational understanding of these formats is critical for any modding endeavor. Key formats include MPQ (archive files), DBC (database client files), M2 (3D models for dynamic objects), WMO (3D models for static world objects), BLP (textures), and Lua/XML (scripting and user interface definitions).[4]

**MPQ (Mo'PaQ) files** serve as the primary archive format for bundling game data. For WoW Classic (Vanilla), MPQ files load in a specific hierarchical order, with Patch-Z.mpq typically holding the highest priority. This loading order is crucial, as it determines which files take precedence when multiple versions exist, allowing custom content to override default game assets.[8] Tools such as MPQ Editor are indispensable for managing these archives, enabling developers to extract existing game files, create new MPQ archives, and modify the contents of existing ones. This functionality is critical for packaging and deploying client-side patches.[9]

**DBC (Database Client) files** contain critical game data that both the client and server rely on, such as spell definitions, item properties, and NPC characteristics. For instance, the spell.dbc file is used by the client for display purposes (e.g., spell names and descriptions) and by the server for calculations (e.g., damage, healing, spell effects, and validating client requests). This dual reliance means that modifications to spells, items, or NPCs typically necessitate updating both the client-side DBCs (via MPQ patches) and the server-side database entries.[12] Specialized DBC editors, such as the WoW Spell Editor [12] or more beginner-friendly alternatives like Trinity Creator [13], can import and export DBC data to and from SQL databases, facilitating easier manipulation and management of this core game data.

**M2 files** (also known as .mdx or .m2) define dynamic 3D model objects, encompassing player characters, creatures, weapons, armor, and various interactive doodads. Each M2 file describes the vertices, faces, materials, texture names, and animations of a model.[4] For WoW Classic, M2 files typically correspond to versions

256-257.[5] It is important to note that M2 files often do not store all animation data internally; they frequently rely on external files such as .anim, .skin, .phys, .bone, and .skel for additional model information, with the specific files varying based on the client version.[4]

**WMO (World Model Object) files** are used for large, static world structures like buildings, dungeons, and other architectural elements that form the game's environment.[6] WMOs possess a chunked structure and support advanced features such as liquids (e.g., water, lava), complex lighting systems, and portal culling, which optimizes rendering by hiding unseen areas.[15]

**Textures** in WoW are primarily stored as .blp (Blizzard Picture) files. For editing, these can be converted to standard image formats like .tga or .png using tools such as BLPNG Converter [11] or various online converters.[17] After modification in image manipulation programs like Photoshop or GIMP, they are converted back to .blp for in-game use. Custom textures intended specifically for Lua addons can often be used directly as .tga files, provided they are placed in designated directories like Interface\Icons within the WoW client folder.[16]

**Lua** is the core scripting language for developing in-game addons and implementing many server-side custom mechanics within private server frameworks.[18] The WoW API exposes a rich set of functions that allow Lua scripts to interact directly with the game world, manipulate the user interface, respond to in-game events, and access player data.[19]

**XML** is used in conjunction with Lua to define the structure and layout of user interface elements within addons.[18]

The extensive review of these file formats reveals that WoW Classic's game data is not a singular, easily modifiable entity but rather a complex web of interconnected files. For example, implementing a new custom item for the "Lionheart Love Patch" might involve not only updating its entry in a DBC file [9] but also creating a new 3D model (M2) and its corresponding texture (BLP) [17], and then packaging all these client-side assets into an MPQ patch.[8] This intricate dependency chain underscores that custom content development is rarely a linear process; instead, it is a complex, iterative cycle demanding familiarity with multiple file types and the specialized tools required to manipulate each. Consequently, developers cannot afford to specialize in just one file type; they require a broad, holistic understanding of the entire WoW Classic data structure and the comprehensive toolchain necessary to modify it. This inherent complexity represents a significant barrier to entry for new modders but simultaneously offers immense potential for deep and creative customization.

### Legal Considerations and Community Best Practices

It is crucial to understand that operating a World of Warcraft private server is not officially sanctioned by Blizzard Entertainment and typically constitutes a breach of their End User License Agreement (EULA).[22] From a legal standpoint, hosting a private server without explicit permission from Blizzard is technically considered copyright infringement due to the unauthorized use and distribution of their intellectual property.[22]

A critical best practice to minimize legal risks is to strictly avoid any form of monetization. Charging players for access, offering paid features (e.g., in-game advantages, cosmetic items for real money), or accepting donations that could be construed as payment is highly likely to attract Blizzard's attention and potential legal action.[22] To further mitigate risks, it is generally recommended to keep the private server small, operate it on an invite-only basis, and strictly adhere to a non-monetized model.[22]

The explicit legal disclaimers regarding private servers highlight that WoW Classic modding operates within an unofficial, legally ambiguous, and somewhat "underground" ecosystem. This unofficial status directly dictates how development is conducted: tools are predominantly community-made [10], technical documentation is often derived from reverse-engineering efforts [4], and content distribution relies heavily on community channels like GitHub [28] and Discord [30] rather than official marketplaces. This environment fosters a highly collaborative and often open-source development culture, but it also implies less stability, a lack of official support, and a greater reliance on shared community knowledge for troubleshooting and tool maintenance. The ultimate success of the "Lionheart Love Patch" will depend not only on the technical prowess of the development team but also on their active engagement with the existing modding community. This includes leveraging shared knowledge bases, participating in troubleshooting discussions, and staying updated on community-developed tools. The project must fully embrace the collaborative, open-source ethos prevalent within the private server scene.

## II. Establishing Your Development Environment

Establishing a robust and efficient development environment is the foundational step for creating the "Lionheart Love Patch." This involves setting up a private World of Warcraft Classic server, configuring its database, ensuring network connectivity, and assembling a comprehensive suite of modding tools.

### Setting Up a Private WoW Classic Server (e.g., VMaNGOS, AzerothCore)

The first critical component is a private WoW Classic server, which will serve as the sandbox for development and testing.

**Hardware and Software Requirements:** For individual developers or small groups, a standard desktop computer equipped with at least 2 CPU cores and 4GB of RAM is generally sufficient. A minimum of 10GB of storage space is specifically required for game data extraction during the setup process.[22] For larger player bases or more complex custom content, considering cloud hosting or a dedicated server with more robust specifications (e.g., more RAM and CPU cores) is highly advisable.[22] WoW servers can operate on both Windows and Linux (e.g., Ubuntu). While many pre-built server "repacks" are Windows-friendly, Linux distributions like Ubuntu 22.04 LTS are commonly used for cloud-based server setups due to their stability and performance.[22] Crucially, the specific version of the WoW Classic client must precisely match the server files; for the original Vanilla/Classic experience, this typically means using the 1.12.1 client with build 5875.[33] The server's backend database requires MySQL version 5.5 or 5.6.[33] The core server software, such as VMaNGOS [33] or AzerothCore [34], forms the backbone of the private server. Pre-built "repacks" can significantly simplify the initial setup by providing pre-compiled server files, thus eliminating the need for manual compilation.[22]

**Core Compilation and Game Data Extraction:** For those opting to compile the server core from source (e.g., VMaNGOS), a Linux environment necessitates the installation of various development dependencies, including g++, cmake, libmysqlclient-dev, openssl, checkinstall, zlib1g-dev, libtbb-dev, and libace-dev.[33] The core source code is typically cloned from a GitHub repository (e.g., git clone -b development https://github.com/vmangos/core), configured using CMake (e.g., with flags like -DSUPPORTED_CLIENT_BUILD=5875 -DUSE_EXTRACTORS=1), and then compiled using the make command.[33]

Game data extraction is a particularly time-consuming process. It involves copying the entire WoW client folder to the server and then executing a series of extractor programs such as mapextractor, vmapextractor, vmap_assembler, and MoveMapGen.[33] This extraction process can take upwards of 20 hours on a single CPU core.[33] Once extracted, the dbc, maps, vmaps, and mmaps folders must be moved to the server's bin directory for the server to function correctly.[33] The research explicitly highlights the game data extraction process as "by far the longest in the whole guide time wise" [33], with a potential duration exceeding 20 hours on a single

CPU core. This substantial initial time investment for establishing the foundational server environment strongly suggests that developers should critically evaluate the trade-offs. While compiling the server core from source offers the highest degree of control and customization, the significant initial overhead could become a major bottleneck for rapid prototyping and development of the "Lionheart Love Patch." For a project as ambitious and feature-rich as the Lionheart Love Patch, where rapid iteration and testing are paramount, strategically prioritizing ease of setup (e.g., utilizing pre-extracted data or pre-built repacks [22]) over a full source compilation for the initial development environment might be a more efficient decision to accelerate the transition to actual content development.

**MySQL Database Configuration and Management:** The installation of MySQL 5.6 on modern Linux distributions like Ubuntu 22.04 requires specific, often manual, steps due to version compatibility challenges.[33] Once MySQL is installed, a dedicated mangos user (with a specified password, e.g., mangos), along with several databases (logs, realmd, characters, mangos), must be created, and appropriate privileges granted to the mangos user.[33] These databases are then populated by importing various SQL files (e.g., logs.sql, logon.sql, characters.sql, and the main world_full_*.sql database file).[33] Subsequent updates are applied through migration scripts to ensure data consistency and currency.[33] The realmlist table within the realmd database must be updated to include the server's chosen name and its public IP address, which players will use to connect.[33] Finally, core server configuration files, such as mangosd.conf and realmd.conf, need to be reviewed and adjusted to ensure correct database connection strings.[33] For performance optimization, especially under heavy loads, increasing MySQL memory limits in its configuration file can help reduce latency and prevent crashes.[22]

The extensive and detailed steps required for MySQL setup and configuration [33] fundamentally underscore that the database is far more than just a backend component; it serves as the central, authoritative repository for virtually all dynamic game content. This includes player data, item definitions, NPC characteristics, and quest logic. The ambitious proposal for the "Lionheart Love Patch," which includes new NPCs (Madame Florence), custom items (heart-shaped cloaks), and intricate questlines (Lost Love of Elyndra), directly translates into a series of critical database modifications. The ability to directly edit tables like item_template, creature_template, and quest_template [13] via SQL commands [9] signifies that the database is the primary interface for implementing and managing the *existence* and *properties* of these new gameplay elements, while Lua handles their *behavior*. Consequently, strong proficiency in SQL and meticulous database management (including regular and automated backups [22]) are not merely beneficial but absolutely paramount. Errors or inconsistencies within the database can lead to severe issues, ranging from corrupted game data to complete server crashes [22], making robust database practices a cornerstone of the project's stability.

**Network Configuration and Client realmlist.WTF Setup:** To allow players to connect, specific network ports must be opened on the server's firewall: port 3724 for authentication and port 8085 for the game world connection.[22] This typically involves configuring a firewall like UFW on Linux. The client's realmlist.WTF file, located in the WoW client folder, must be edited to point to the server's public IP address (e.g., set realmlist x.x.x.x).[22] For local development and testing on the same machine, the loopback address 127.0.0.1 is used.[35] Common connection issues frequently stem from incorrect IP addresses in realmlist.WTF, firewall blocks (e.g., antivirus software [35]), or the server processes (realmd, mangosd) not running correctly.[22]

While seemingly a minor detail, network configuration (including proper port forwarding and the correct realmlist.WTF entry) [33] is a surprisingly common and critical point of failure in private server setup. The success of playtesting, particularly involving a wider community, hinges entirely on players' ability to reliably connect to the server. A single misconfigured port or an active firewall rule can prevent the entire "Lionheart Love Patch" from being accessible for external testing, rendering development efforts in isolation futile. Therefore, thorough and systematic testing of network connectivity should be conducted from the very outset of the project, not merely as an afterthought once content development is underway. This also necessitates providing clear, concise, and easy-to-follow instructions for testers on how to modify their realmlist.WTF file and common troubleshooting steps for connection issues.

### Essential Modding Tools for WoW Classic

The development of the "Lionheart Love Patch" will rely on a diverse array of specialized tools, many of which are community-developed and open-source.

**MPQ Editors:** Tools like MPQ Editor [10] are indispensable for managing .mpq archives. They allow developers to extract existing game files, create new MPQ archives, and modify the contents of existing ones. This functionality is critical for packaging and deploying client-side custom content, such as new models, textures, or UI elements.[8]

**DBC Editors:** The WoW Spell Editor [12] is a prime example of a specialized DBC editor. It facilitates the import and export of DBC files to and from SQL databases, enabling straightforward editing of spells, items, NPCs, and quests.[13] Trinity Creator is also noted as a user-friendly program for these specific tasks.[13]

**WoW Model Viewer (WMV):** An open-source application that allows users to view and export all in-game models—characters, armor, weapons, monsters, and spell graphics—outside of the game environment.[14] WMV supports multiple WoW client versions, including Classic [23], and is frequently used by machinima artists for "blue- or green-screen-effect" recordings.[23]

**Blender with WoW Addons:** Blender, a powerful open-source 3D creation suite, becomes a central hub for asset development when augmented with specific WoW-focused add-ons.

- **WoWbjectImporter:** This Blender add-on is designed specifically for *importing* M2 and WMO models into Blender. It automates the application of game textures and materials, and sets up UV animations.[25] It requires models to be exported from wow.export.[25] It is important to note that WoWbjectImporter is an importer, not an exporter, meaning it does not facilitate exporting custom models back into WoW formats.[25]
- **Blender WoW Studio:** This is a more recent continuation of older WMO import/export scripts.[15] It enables the opening and editing of WMOs from WoW cities directly within Blender.[26] It also supports M2 file import.[26] This addon is designed for recent Blender versions and is compatible with WoW Lich King Client 3.3.5 and Turtle WoW Client Files.[26] Its explicit export capability for M2/WMO files is not clearly confirmed in the provided information.[26]
- **M2Mod Redux:** This tool is primarily used for creating model edits for WoW. It can extract character files with their geosets, making them more suitable for modifications or creating "universal modular models".[14] It is also mentioned for converting M2 files to an M2I format, which can then be imported into Blender.[37]

**Texture Converters:** Tools like BLPNG Converter [11] are essential for converting textures between WoW's proprietary .blp format and standard image formats like .tga and .png.[16] Various online converters also exist for this purpose.[17]

**Particle Effect Tools:** While WoW Classic has an in-game console variable (graphicsParticleDensity) to adjust particle density [38], dedicated external tools like Effekseer [39] are available for creating sophisticated particle effects. Effekseer is open-source, multi-platform, and can export effects for various game engines.

The extensive array of tools identified (MPQ Editors, DBC Editors, Model Viewers, Blender addons, texture converters, spell editors) indicates that there is no single, official, or comprehensive modding suite for WoW Classic. Instead, the modding landscape is characterized by a fragmented ecosystem of community-developed, often open-source, and highly specialized tools. Many of these tools are designed for very specific tasks (e.g., WoWbjectImporter is solely for import [25]; WoW Spell Editor focuses on DBCs [12]), necessitating that developers stitch together a complex, multi-stage workflow. The "Lionheart Love Patch" development team will need to dedicate significant time and effort to understanding, installing, and integrating these disparate tools into a cohesive workflow. Compatibility issues between different tools (e.g., specific Blender versions required, nuances of various export/import formats) are highly probable, requiring persistent troubleshooting and active engagement with community forums [15] for solutions and workarounds.

| **Tool Category** | **Specific Tools (Examples)** | **Primary Function** |
|---|---|---|
| **Asset Creation & Management** | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | 1. **Blender** (with Blender Kit, HardOps, Boxcutter, Decal Machine, etc.) | 3D modeling, sculpting, animation, rendering, compositing, video editing, motion tracking, visual effects, game design, etc. |
| | 2. **Blender (with BlenderKit, HardOps, Boxcutter, Decorex, etc.)** | |
| | | |
| | | |
| | | |

| | | |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| | | |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | 2. **Blumix** |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Table 2: Tools for Wow, as well as descriptions, categories, and compatibility.

| Tool Category | Tool Name | Key Functionality | Compatibility |
|---|---|---|---|
| Database Management | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| * * * * | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | 2. | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | \hline \end{tabular} \caption{ \label{tab: 1} | |

```
\begin{tabular}[t]{p{0.99\textwidth}} \end{tabular} } \end{IEEEeqnarray}
\end{IEEEeqnarray} \end{IEEEeqnarray}
\end{IEEEwraptext}
\begin{table*}[t]
\centering
\caption{S_t_s: \label{tab: 1}
\begin{center}
\begin{tabularx}{1.0\textwidth}{ L{0.1\textwidth} L{0.15\textwidth}L{0.7\textwidth}}
\begin{tabular}{ |c|c|c| } \hline
\bf{Category} & \bf{Tool} & \bf{Functionality} \ \hline
\multirow{2}{}{General Purpose} & \multirow{2}{}{-}-& \multirow{2}{}{-}\
\end{longtable}
\begin{longtable}[c]{@{} l l l @{}}
\toprule \end{tabular}
\end{longtable}
\end{table}
\begin{longtable}{ | c | c | c | c | } \hline
\textbf{Category} & \textbf{Tool} & \textbf{Key Feature} & \textbf{Notes} \
\hline
\text{ } & \text{ } & \text{ } & \text{ } \
\textbf{I\text{.}nstallation} & \text{ } & \text{ } & \text{ } \
\end{longtable}
\begin{longtable}[c]{@{} l l l l @{}} \toprule \bf{No.} & \bf{Name} & \textbf{Description} &
\textbf{Used for} \ \hline
\end{tabular}
\end{longtable}
\begin{table}[H]
\begin{tabular}{|c|l|l|l|}
\hline
\multicolumn{1}{|c|}{\textbf{Category}} & \multicolumn{1}{c|}{\textbf{Tool}} &
\multicolumn{1}{c|}{\textbf{Key Functionality}} & \multicolumn{1}{c}{\textbf{Notes}} \
\hline
\addlinespBace{0.5ex}
\textbf{Server Management} & \multirow{2}{}{-}- & \multirow{2}{}{-}- & \multirow{2}{}{-}- \
\addlinespace{0.2em}
\multirow{2}{}{- } & \multirow{2}{}{-} & \multirow{2}{}{-} & \multirow{2}{*}{-} \
\addlinespace{0.2em}
& \multicolumn{1}{c}{}\textbf{H} & \multicolumn{1}{c}{}\textbf{U} &
\multicolumn{1}{c}{}\textbf{V} \ \hline
\end{tabular}
\end{enumerate}
\end{enumerate}
\end{document}
```

## Referanser

1. World of Warcraft Classic - Wikipedia, brukt juni 3, 2025, https://en.wikipedia.org/wiki/World_of_Warcraft_Classic
2. Classic - Warcraft Wiki - Your wiki guide to the World of Warcraft, brukt juni 3, 2025, https://warcraft.wiki.gg/wiki/Classic
3. World of Warcraft: Classic, brukt juni 3, 2025, https://warcraft.wiki.gg/wiki/World_of_Warcraft:_Classic
4. WowDev wiki, brukt juni 3, 2025, https://wowdev.wiki/Main_Page
5. M2 - wowdev, brukt juni 3, 2025, https://wowdev.wiki/M2
6. WMO - wowdev, brukt juni 3, 2025, https://wowdev.wiki/WMO
7. MPQ - wowdev, brukt juni 3, 2025, https://wowdev.wiki/MPQ
8. Any info on WoW client MPQ files? Trying to learn AddOn development : r/wowservers - Reddit, brukt juni 3, 2025, https://www.reddit.com/r/wowservers/comments/1210clb/any_info_on_wow_client_mpq_files_trying_to_learn/
9. WoW Server Development: Patching Custom Items the Fast Way - YouTube, brukt juni 3, 2025, https://m.youtube.com/watch?v=b5ponz4RXW8

10. MPQ Archives - Downloads, brukt juni 3, 2025, http://www.zezula.net/en/mpq/download.html
11. How to change the WoW log in screen, music and clean it up - Warmane | Forum, brukt juni 3, 2025, https://forum.warmane.com/showthread.php?t=468497
12. stoneharry/WoW-Spell-Editor: A Spell Editor for WoW ... - GitHub, brukt juni 3, 2025, https://github.com/stoneharry/WoW-Spell-Editor
13. Modding tool and tutorial recommendations for creating a custom server? : r/wowservers, brukt juni 3, 2025, https://www.reddit.com/r/wowservers/comments/1irnbwr/modding_tool_and_tutorial_recommendations_for/
14. WoW data extraction resource sheet - Steak's Docs, brukt juni 3, 2025, https://thunderysteak.github.io/wow-data-extract-cheat-sheet
15. WowDevTools/Blender-WMO-import-export-scripts: Script addon for Blender featuring World of Warcraft World Model Object (WMO) editing functionality. - GitHub, brukt juni 3, 2025, https://github.com/WowDevTools/Blender-WMO-import-export-scripts
16. Adding Custom Textures In-Game - WoWInterface, brukt juni 3, 2025, https://www.wowinterface.com/forums/showthread.php?t=50495
17. Create and Use a Custom Texture with WeakAuras : r/wow - Reddit, brukt juni 3, 2025, https://www.reddit.com/r/wow/comments/lc471f/create_and_use_a_custom_texture_with_weakauras/
18. How do I learn making WoW mods? - WoWInterface, brukt juni 3, 2025, https://wowinterface.com/forums/showthread.php?t=4138
19. Lua functions - Warcraft Wiki - Your wiki guide to the World of Warcraft, brukt juni 3, 2025, https://warcraft.wiki.gg/wiki/Lua_functions
20. Lua Coding Tips - Warcraft Wiki - Your wiki guide to the World of ..., brukt juni 3, 2025, https://warcraft.wiki.gg/wiki/Lua_Coding_Tips
21. Adding Custom Models - Modding Tutorials - WordPress.com, brukt juni 3, 2025, https://xylozi.wordpress.com/eu4/eu4-adding-custom-models/
22. Setting Up a World of Warcraft Server [Step-by-Step Guide] - HostGator, brukt juni 3, 2025, https://www.hostgator.com/blog/setup-world-of-warcraft-server/
23. WoW Model Viewer - Warcraft Wiki, brukt juni 3, 2025, https://warcraft.wiki.gg/wiki/WoW_Model_Viewer
24. Wow Model Viewer |, brukt juni 3, 2025, https://wowmodelviewer.net/
25. ThatAsherGuy/WoWbjectImporter: A World of Warcraft Model Importer for Blender - GitHub, brukt juni 3, 2025, https://github.com/ThatAsherGuy/WoWbjectImporter
26. Open WMO's with Blender - Turtle WoW - Forums, brukt juni 3, 2025, https://forum.turtle-wow.org/viewtopic.php?t=14236
27. Community Projects - wowdev, brukt juni 3, 2025, https://wowdev.wiki/Community_Projects
28. How do I compile a mod from github? : r/feedthebeast - Reddit, brukt juni 3, 2025, https://www.reddit.com/r/feedthebeast/comments/1aoyo0x/how_do_i_compile_a_mod_from_github/

29. modding-games · GitHub Topics · GitHub, brukt juni 3, 2025, https://github.com/topics/modding-games
30. Am I the only one irritated with the role of Discord in modding? : r/FO4mods - Reddit, brukt juni 3, 2025, https://www.reddit.com/r/FO4mods/comments/1i6vo42/am_i_the_only_one_irritated_with_the_role_of/
31. Community Feedback: Discord Server : r/LostRecordsGame - Reddit, brukt juni 3, 2025, https://www.reddit.com/r/LostRecordsGame/comments/1iwbhg2/community_feedback_discord_server/
32. Am I the only one irritated with the role of Discord in modding? : r ..., brukt juni 3, 2025, https://www.reddit.com/r/FO4mods/comments/1i6vo42/am_i_only_one_irritated_with_the_role_of/
33. Creating and Hosting A World of Warcraft Private Server - Rider ..., brukt juni 3, 2025, https://riderjensen.com/blog/create-wow-server
34. Creating a small private server : r/wowservers - Reddit, brukt juni 3, 2025, https://www.reddit.com/r/wowservers/comments/1fs01km/creating_a_small_private_server/
35. Super Easy guide to create your own WoW private server (2023) WOTLK Edition - YouTube, brukt juni 3, 2025, https://www.youtube.com/watch?v=NdRK5WWfYp0
36. 3D WoW Machinima Guide [Easy Animation] - YouTube, brukt juni 3, 2025, https://www.youtube.com/watch?v=TsQ-wp_Towc
37. importing World of Warcraft models into Blender - Other Software, brukt juni 3, 2025, https://blenderartists.org/t/importing-world-of-warcraft-models-into-blender/526693
38. CVar graphicsParticleDensity - Warcraft Wiki, brukt juni 3, 2025, https://warcraft.wiki.gg/wiki/CVar_graphicsParticleDensity
39. Particle Effect Creation tool - Effekseer, brukt juni 3, 2025, https://effekseer.github.io/en/