Guide contents

Set up the Toolset and protect the project

Build a tiny playable loop

Keep custom resources and dependencies clear

Sources / References

Documentation: In progress **Quick start: In progress**

Offline copy from ModDocs. Source links remain clickable.

Neverwinter Nights modding

Neverwinter Nights uses the **Aurora Toolset** to create playable modules containing areas, creatures, dialogue, scripts and encounters. This guide concentrates on Neverwinter Nights: Enhanced Edition (NWN:EE); older original-game tutorials can be useful, but packaging and tool behavior should be checked in the edition you run. The NWN Wiki Toolset manual is the starting reference.

> **Quick start:** create a new module, add one small area with a safe player start and a visible object, save it, then launch it in the game. Only after that add a simple dialogue or script.

Set up the Toolset and protect the project

On Windows, the NWN:EE Toolset is an older 32-bit application located under the game's `bin/win32` folder according to the manual. Open it from your installation. If icons disappear at unusual Windows display scaling, the manual describes trying 100%, 150% or 200% scaling or an application-level DPI override. Save often and keep a copy of the module outside the game's working directory; a toolset crash can damage work in progress.

The module is stored as a `.mod` file. The Toolset has a **Create a new module** option, File > New, and Ctrl+N. Name the module distinctly from any sample or downloaded module. Create an area with the creation wizard, then use the area editor to place a

start location and a simple object. A module that opens in the Toolset but lacks a usable start point may fail as a playable test.

Build a tiny playable loop

Start with one room. Add a door or transition only after the starting area runs. Place a creature or item whose presence is easy to verify. Save the module, close the Toolset if needed, and launch the game to load the module. Walk around, interact with the object and confirm the player can leave any transition you added. Repeat after each change. The Toolset topic directory links the area and module-specific pages.

Once the room works, add a conversation with one line of text. For scripted behavior, attach one short NWScript event that changes a visible state or emits a diagnostic message. Compile it in the Toolset and test the exact event trigger in-game. A script existing in the module does not prove it is attached to the correct object or event. Keep a note of the object, event slot and expected outcome so you can distinguish a compile problem from a trigger problem.

Keep custom resources and dependencies clear

Custom models, textures, sound and rules resources may require additional packages such as HAK files. Add these only when the basic module runs. Maintain a list of every external resource and version, then test on a clean NWN:EE user directory with no other module overrides. The Toolset manual documents the `-userdirectory` option, which can help isolate project-specific files. Avoid distributing assets copied from the base game.

If the Toolset's F9 test launch behaves unexpectedly, use another launch path; the manual notes this can happen. If the Toolset crashes, look for its `.BackupMod` or temporary module folder, but copy them before attempting recovery. Version the source module and test a released `.mod` separately from your working copy.

This guide is **unfinished**: it explains a first playable module and a staged content path, but it does not yet include a complete NWScript quest, HAK packaging or multiplayer persistent-world deployment walkthrough.

Sources / References

External project and documentation links cited in this guide. Check the upstream release and version before following version-sensitive steps.

- NWN Wiki Toolset manual