

Neubercart Advanced Help Editor

tl;dr

The **Neubercart Advanced Help Editor Module** (`nc_advanced_help_editor`) is a specialized Drupal 7 extension intended to extend and enhance the Advanced Help Module. It provides an endpoint for the management of Advanced Help implementations. It helps site builders to investigate Advanced Help availability on a module-by-module basis, audits directory and file ownership and permissions, and offers specialized tools (`unlock-help.bash` & `lock-help.bash`) for securely implementing and maintaining Advanced Help of any given module. A built-in WYSIWYG editor enables the creation and maintenance of static `.html` help files and `.ini` configuration file used to control the Advanced Help for any given module. Together, these features help to transform the Advanced Help production process from a command line oriented, developer dependent manual process into a graphical experience that site builders can engage in to more easily produce and maintain high-quality, comprehensive Advanced Help documentation for modules.

Purpose

The purpose of the **Neubercart Advanced Help Editor Module** (`nc_advanced_help_editor`) is to transform the module documentation process from something that is currently dependent on Code Developers to something that can be more actively engaged in by Documentation Developers, Site Builders, and even Administrators. Traditionally, Advanced Help has been locked inside static `/help` directories, requiring Code Developers to hand-craft `.html` files and `.ini` configurations. This approach excluded non-developers from contributing meaningfully to module documentation. By introducing a dedicated administrative endpoint, integrated WYSIWYG editing, and secure command-line utilities, `nc_advanced_help_editor` democratizes the documentation workflow. In essence, `nc_advanced_help_editor` converts Advanced Help into a collaborative subsystem that reflects the needs of the entire Drupal ecosystem rather than just the agenda of its coders.

- For Documentation Developers: It provides a structured, graphical environment in which to author and maintain Advanced Help topics without the need to work on the command line. The only command line interaction needed is to run the scripts that unlock and lock the Advanced Help `/help` subdirectories in order to maintain them.
- For Site Builders: It offers visibility into the modules they maintain and offer with respect to whether those modules offer more than the minimum (`README.txt`). For those modules they wish to offer Advanced Help for, a complete development environment is offered to help them curate extended documentation as part of module deployment and site assembly.
- For Site Administrators: It ensures proper ownership, permissions, and security of help directories, while giving them oversight through audit tools and locking/unlocking mechanisms, and visibility into the availability of Advanced Help for any given module – which may help to spur along its development as people come to understand that a form of dynamic, extendable, correctable information regarding the purpose, capabilities, configurability and functionality of modules is available and possible from within the system.

Scope

In Scope

- Endpoint:: Creation of an endpoint in the Drupal system to make the Advanced Help Editor subsystem available via direct URL.
- Menu: Creation of a menu item in the administrative toolbar to make accessing the Advanced Help Editor via the GUI.
- Manifesting: List all installed modules along with their Advanced Help status, which is essentially the existence of an Advanced Help sitelet or just a README.txt file.
- Audit: Enumerate and offer guidance regarding the directory and file system ownership and permissions with respect to the Advanced Help sitelet directory (/help) and related files.
- CLI Utilities: Provide `unlock-help.bash` and `lock-help.bash` command line scripts to facilitate the file system alterations necessary to enable interested parties to develop the Advanced Help sitelet for a given module, and then secure it afterwards.
- Editing: Provide a graphical Advanced Help editor that enables interested parties to create both the `.ini` configuration file that governs Advanced Help, as well as the `.html` files that constitute the actual content of the Advanced Help sitelet.
- Oversight: Functionality that enables Administrators to track changes, enforce security, and ensure consistency across module Advanced Help sitelets.

Out of Scope

- Independence: This module does not intend to replace or modify the built-in Help system. It is intended to establish, modify and/or replace Advanced Help sitelets for core and contributed modules.
- Specialization: This module is not intended for the creation, modification or deletion of general site content, user guides, or documentation unrelated to Advanced Help /help sitelets.
- Focus: This module is not intended for the creation, modification or deletion of anything other than /help subdirectories and the static .ini files and static .html files contained within them. It is not intended to help develop interactive tutorials, live tours, or dynamic walkthroughs, nor to alter the programming or functionality of modules.
- Integrations: This module is not intended to integrate or connect with external documentation platforms or otherwise synchronize itself with outside parties or services other than via the platform update server, which will signal the release of updated Advanced Help documentation in the form of a module update.
- Translation: This module is not intended to provide translation services. While Advanced Help does accommodate translated versions of Advanced Help content, the translation process shall remain manual.

Benefits

- Democratization: This module shifts Advanced Help from a developer-only subsystem into a collaborative environment where Documentation Developers, Site Builders, and Administrators can all contribute meaningfully.
- Centralization: This module provides a single administrative endpoint to oversee, audit, and maintain Advanced Help subsystems across all installed modules, reducing fragmentation and guesswork.
- Security: This module introduces specialized lock/unlock command-line tools and permission audits, ensuring that help directories remain secure while still accessible for authorized editing.
- Usability: This module offers a WYSIWYG editor for creating and maintaining .html and .ini files, eliminating the need for manual file editing and lowering the barrier for non-developers.
- Visibility: This module surfaces whether modules provide full Advanced Help subsystems or only README files, helping site builders identify gaps and prioritize documentation efforts.
- Quality: This module ensures that help content is structured, correctable, and extendable, improving the overall reliability and usefulness of module documentation.

Intended Outcomes

- Cultural: This module is intended to help establish a system where Documentation Developers, Site Builders, and Administrators can all participate in creating and maintaining Advanced Help, reducing reliance on code developers alone.
- Transparency: This module is intended to help provide clear visibility into which modules ship with Advanced Help subsystems, which rely only on README files, and where documentation gaps exist.
- Security: This module is intended to help secure and control editing: Ensure that help directories and files can be safely unlocked for editing and then locked again, maintaining strong ownership and permission controls.
- Efficiency: This module is intended to help streamline the authoring process: Deliver a WYSIWYG editor and administrative endpoint that simplify the creation of .html and .ini files, lowering the barrier for non-developers to contribute.
- Experience: This module is intended to facilitate higher documentation quality: Produce structured, correctable, and extendable help content that improves usability, reliability, and the overall Drupal ecosystem experience.

Requirements

Functional Requirements

- Interface: Provide a dedicated endpoint in the Drupal administrative toolbar for managing Advanced Help subsystems, including listing modules and their help status.
- Detection: Automatically identify whether installed modules contain an Advanced Help subsystem (/help directory) or only a README file, and report this in the admin interface.
- Audit: Display and verify file system ownership and permissions for /help directories and their contained .html and .ini files.
- CLI: Provide `unlock-help.bash` and `lock-help.bash` scripts to securely manage write access to module help directories.
- Editor: Provide a WYSIWYG graphical editor for the production and maintenance of the master .ini configuration file and associated static .html Advanced Help files, thus lowering the participation barrier for non developers.
- Reporting: Generate reports on the disposition of Advanced Help subsystems, including availability, ownership, permissions, and completeness.
- Preview: Allow editors to preview how Advanced Help content will render within the Advanced Help system prior to committing changes.

Non-Functional Requirements

- Performance: The module must operate efficiently, ensuring that administrative reporting and WYSIWYG editing do not introduce noticeable delays or degrade overall site performance.
- Security: All interactions with /help directories must respect Drupal's security model, enforcing strict ownership and permission checks, and ensuring that lock/unlock operations cannot be bypassed.
- Usability: The administrative interface and WYSIWYG editor must be intuitive, accessible, and consistent with Drupal's UI patterns, lowering the barrier for non-developers.
- Maintainability: The module's codebase should be modular, well-documented, and easy to extend, allowing future enhancements without major refactoring.
- Reliability: The system must handle errors gracefully, provide clear feedback to users, and ensure that help content is not corrupted during editing or locking/unlocking operations.
- Compatibility: The module must integrate seamlessly with Drupal 7 core and contributed modules, without requiring changes to existing Advanced Help subsystems.
- Auditability: All changes to help files should be logged, enabling administrators to track edits and maintain accountability.

Dependencies

- Core: Requires the core framework, including the administrative toolbar and file system APIs, to provide integration points for Advanced Help management.
- Advanced Help module: Depends on the existing Advanced Help subsystem as the foundation for /help directories, .html files, and .ini configurations.
- .ini File Editor: CodeMirror
- .html File Editor: Jodit

Permissions

- Custom Permission: This module introduces a specialized permission, the “Manage Advanced Help Documentation” permission, that is intended to enable Documentation Developers access to the Advanced Help Editor while preventing them from wider system-level access.
 - Documentation Developers: Granted this custom permission, enabling them to create and edit .html and .ini help files without having to be granted too much in terms of administrative rights.
 - Administrators: Inherit the permission automatically, ensuring they retain full control over documentation management alongside broader site responsibilities.
 - Site Builders: Also inherit the permission through their administrative role, allowing them to configure and maintain the help system.

Installation

- Download and place module files: Download and place the module archive in the most appropriate spot in your file system. Commonly used locations include:
 - /sites/all/modules
 - /sites/all/modules/ custom
- Expand the module archive to expose the source files
 - `tar -zxvf <archive_name>.tgz`
- In the Administrative GUI, navigate to Administration → Modules.
- Search for Neubercart Anniversary Discount Coupon in the list.
- Check the box to enable it
- Click Save configuration.
- Clear all caches
- Confirm the following menu items appear in the Administration toolbar:
 -
 -
 -
 -

Configuration

- This module does not require configuration aside from assigning the appropriate Permissions to Documentation Developers

Usage

-
-
-
-
-

Screenshots

Potential Extensions

-
-
-
-