



Neverwinter Nights 2 Legends Info Plugin

Version 2.5.1 EE64
October 6, 2026
Marshall Vyper
© 2026 Marshall Vyper

Document Revision Information

Revision	Date	Author	Description
1.30	December 19, 2010	Marshall Vyper	Initial Release
1.31	December 21, 2010	Marshall Vyper	Added Town Crier
1.40	August 17, 2011	Marshall Vyper	Added several new vars
1.42	September 25, 2011	Marshall Vyper	Changes to Plugin Tool
1.81	October 17, 2012	Marshall Vyper	Skill Checks on Triggers
1.82	October 23, 2012	Marshall Vyper	Quest based triggers
1.86	December 17, 2012	Marshall Vyper	Visuals, Animations, New Info Types
2.0	December 30, 2016	Marshall Vyper	Update to 2.0 doc
2.5.1	October 6, 2026	Marshall Vyper	Manual updated for 2.5.1 EE64.

Plugin Revision Information

Revision	Date	Author	Description
1.3	December 19, 2010	Marshall Vyper	Initial Release
1.4	August 17, 2011	Marshall Vyper	Added new variables, New Plugin Systems
1.41	September 14, 2011	Marshall Vyper	Minor Enhancements
1.42	September 25, 2011	Marshall Vyper	Changes to Plugin Tools
1.5	October 6, 2011	Marshall Vyper	Update INI code and GUI fixes.
1.8	September 15, 2012	Marshall Vyper	Numerous Bug Fixes Various Enhancements (See Upgrade Doc)
1.81	October 17, 2012	Marshall Vyper	Bug fixes and new skill checking on triggers.
1.82	October 23, 2012	Marshall Vyper	Quest Based triggers
1.86	December 17, 2012	Marshall Vyper	Visuals, Animations, New Info Types
1.88	March 2, 2013	Marshall Vyper	Updated to current codebase. Added extra visual effects.
1.90	July 18, 2013	Marshall Vyper	Bug Fixes, see Upgrade Doc
1.92	December 23, 2013	Marshall Vyper	Bug fix for color names.
1.93	July 17, 2015	Marshall Vyper	Enhancements, see upgrade doc
2.0	December 30, 2016	Marshall Vyper	New 2.0 Features see upgrade doc.
2.5.1	October 6, 2026	Marshall Vyper	EE64 and current Connector/NET compatibility.

Table of Contents

TABLE OF CONTENTS	3
INTRODUCTION	4
FEATURES.....	4
LEGENDS PLUGINS	5
<i>What's a Plugin</i>	<i>5</i>
<i>Important To Know.....</i>	<i>5</i>
PART 1 – GETTING STARTED.....	6
<i>What's in the File</i>	<i>6</i>
<i>YouTube Video Walkthroughs</i>	<i>7</i>
<i>Installation.....</i>	<i>7</i>
PART 2 – MAKING IT WORK.....	12
<i>Tutorial 1 – Talk to me Goose.....</i>	<i>12</i>
<i>Tutorial 2 – Is it Just Me?</i>	<i>13</i>
<i>Tutorial 3 – Taking Control.....</i>	<i>14</i>
<i>Tutorial 4 – Various Checks.....</i>	<i>15</i>
<i>Tutorial 5 – Other Triggers</i>	<i>17</i>
<i>Tutorial 6 – Making Progress</i>	<i>19</i>
<i>Tutorial 7 – Skilled Progress.....</i>	<i>23</i>
<i>Tutorial 8 – Lights, Camera, Action!.....</i>	<i>25</i>
<i>Tutorial 9 – Hear ye, Hear ye!</i>	<i>28</i>
FINAL SUMMARY	31

Introduction

Sometimes, communicating important events or notable activities in Neverwinter Nights 2 Chat box is just not enough to let your adventurers know they've noticed or accomplished something. Even the Floaty Text above their heads can sometimes be missed. This Legends Plugin helps with this issue. Having trouble making small cut scenes to help relay information to adventurers about their surroundings? Perhaps you are looking to make more use of character skills like Survival, Search and more. All of these can be accomplished with this plugin.

Leveraged by several other Legends Plugins, the Info scripts are simply some customized GUI screens and code that allow you to display various activities on the screen as well as offering the ability to have adventurers perform time sinks when doing various tasks including support for skill checking.

There are four main features of the Info Plugin; the first being a nice display of custom text that can appear on your adventurer's screen, bright, big and noticeable. The second is a progress meter that can be applied to almost any activity from searching for things to using objects with optional skill check, the third is the ability to make simple trigger based cut-scenes and the fourth offers you the ability to create town criers to share any information with nearby players that you desire live (such as when the Goblin Chief is killed or when the town vaults have been looted!)

The Legends Info plugin can be used in stand-alone modules as well as persistent worlds. Without a database, however, the crier functions will not work.

Features

There are many various combinations of things you can do with the Info Plugin and it was deemed useful enough to act as a standalone system. Some of the features this plugin can do for you are listed below:

- Easy to use Configuration Application!
- Display large bright text in the center of the screen.
- Can be used on triggers, objects, just about anything.
- Display progress meters with configurable time intervals with skill checks
- Set persistent information boxes so adventurers only see them once.
- Create simple camera cut scenes that are repeatable or non-repeatable.
- Display information or cut scenes only at certain times of the day.
- Have a town crier speak any messages you care to inject – live!
- Easy to learn and use with step-by-step tutorials.
- Minimal script set – easy installation into any existing world with VERY minor additions to existing scripts.
- You don't need to overhaul what you already have! Just build on it!

Legends Plugins

What's a Plugin

The Legends plugin system is a framework of scripts and Toolset tools originally developed for the Legends Persistent World. Each package includes the shared Master components it requires, so Master does not need a separate download. The Info package includes the Master DLL, HAK and shared scripts, together with the Info resources and documentation. Installation requires only documented configuration and script changes.

Important To Know

Legends plugins share a standardized Master script set. Shared resources may already be present when another Legends plugin is installed. Keep one compatible shared set and confirm resource versions before overwriting existing Master or NWNX scripts.

- ⚠ Legends 2.5 plugin releases have their own patch versions. This package supplies Info 2.5.1 EE64 and Master 2.5.1 EE64. Do not assume every installed plugin has the same version or downgrade newer accepted shared components. Review the supplied Upgrade Guide and existing resource versions before replacing files.

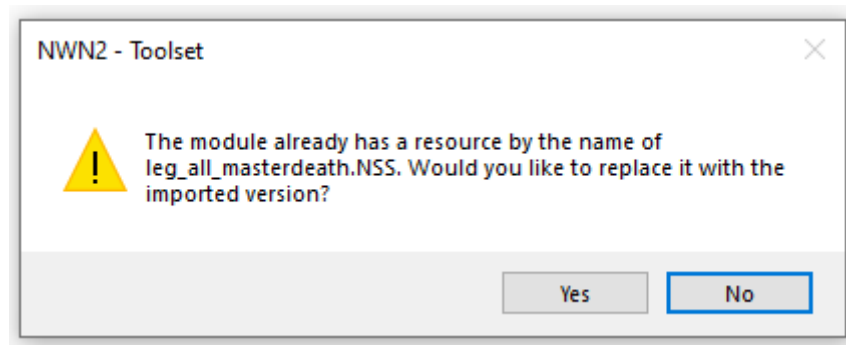


Figure 1 - Master Script Overwrite

Part 1 – Getting Started

What's in the File

The Legends Info 2.5.1 EE64 package contains exactly seven files, described below:

leg_info.erf	The complete builder resource set: 37 resources consisting of 34 scripts, 1 static camera and 2 triggers. It includes the shared Master and NWNX scripts used by Legends 2.5. The Info blueprints appear under the Legends Info category. The tutorials explain their use.
leg_info.dll	This is the NWN2 Toolset plugin that allows you to manage and configure the Legends Info Plugin.
leg_master.dll	This is the NWN2 Toolset plugin that allows you to manage and configure all other Legends plugins and is included in all plugin packages.
leg_info.hak	A hakpak file that contains the custom GUI screens for info and progress bars.
leg_master.hak	This is the master hak pak file that several Legends plugins leverage resources from. They are kept here to avoid duplication of resources in several plugins that share them. It is included in all plugin packages.
leg_info_manual.pdf	This manual, which includes the tutorials and how-to use this info plugin.
leg_info_upgrade.pdf	This file contains any notes and important information you should be aware of BEFORE you upgrade or install a plugin when you already have a pre-existing plugin that is in use.

YouTube Video Walkthroughs

All Legends Plugins at the time of release have video walkthroughs that will assist you in the installation, configuration and typical use of each Legends Plugin. Visit the channel <http://www.youtube.com/user/DJMarshallV> to see these video series. They are detailed and can answer many questions.

Installation

To install this system, you must perform several tasks. You can install this into your existing persistent world or if you just wish to try it out, make a brand new module and follow these steps.

1. Use Neverwinter Nights 2 Enhanced Edition 64-bit Toolset and .NET Framework 4.7.2. The Info database configuration also requires machine-installed Oracle MySQL Connector/NET 8.0.33.1, which supplies MySql.Data 8.0.33.0.
 - Install or confirm these prerequisites on your machine before loading the Toolset plugins.
 - Connector/NET and its support assemblies are not included in the Info package.
2. Close the Toolset. Copy leg_info.dll and leg_master.dll into <NWN2 EE install>\NWN2Toolset\Plugins64. Place the supplied leg_info.hak and leg_master.hak unchanged in Documents\Neverwinter Nights 2\hak, and leg_info.erf somewhere you can easily import it. Keep both PDF guides with the package. This release uses manual installation; setup.exe is not included.
3. Once the files are in place, open the Toolset and load a new or existing module. Select Plugins from the top menu and confirm that Legends Info Plugin and Legends # Master Configurator # are available.

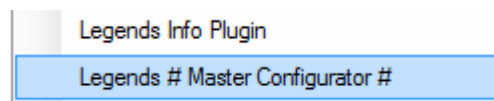


Figure 2 - Plugins Menu

- ⚠ If you do not see these entries, then select View -> Options from the top menu. On the general tab, ensure that “Allow Plugins” is set to “Load all Plugins”

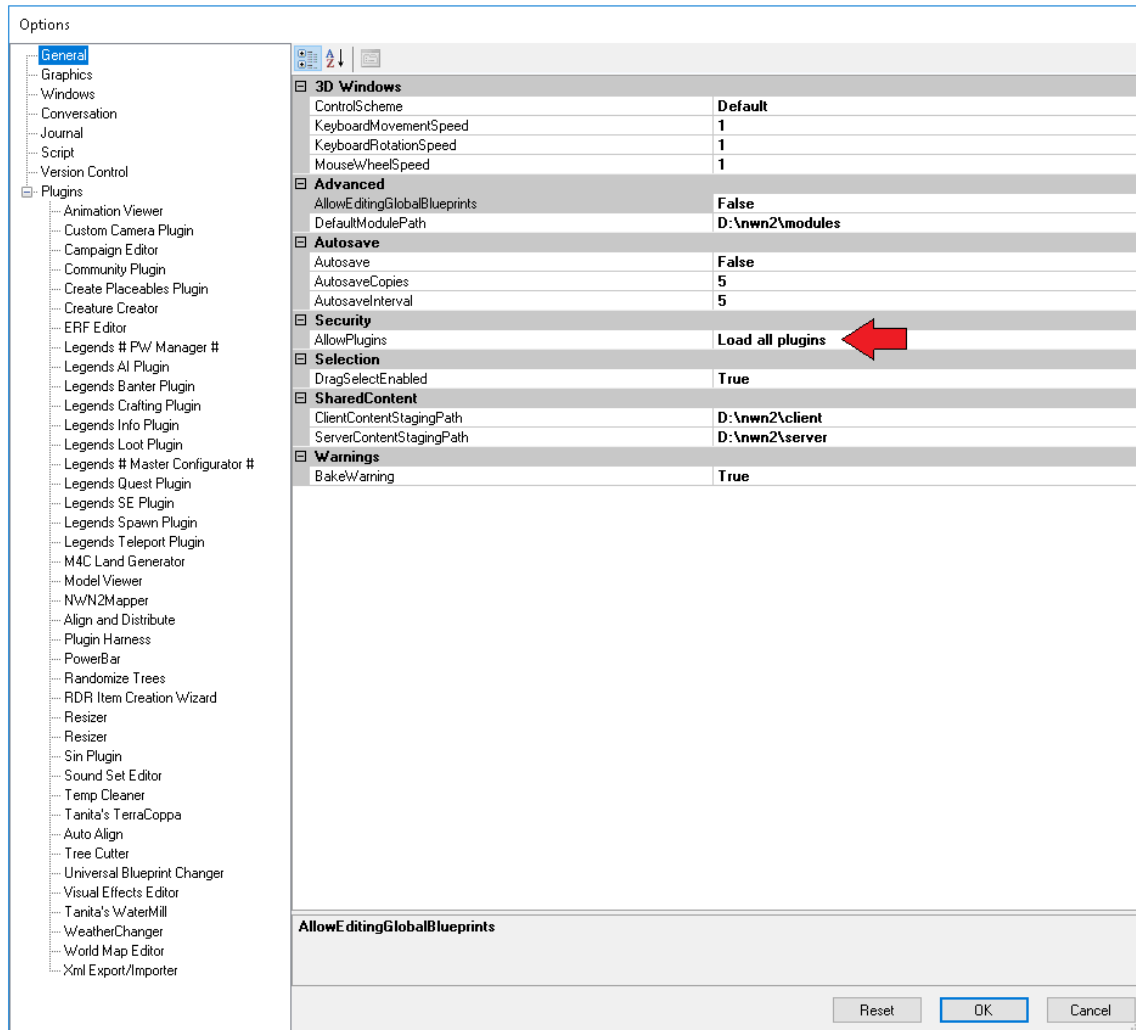


Figure 3 - Security Settings

4. From the File menu, choose Import and select leg_info.erf from wherever you decided to keep it. Import the complete 37-resource set. Existing shared Master or NWNX scripts may produce overwrite prompts like the notice in Figure 1. Replace older included resources with the approved package versions but preserve any newly accepted shared components.
5. If this is your first Legends plugin, then you will need to enter some default information, otherwise, you will simply need to activate your new Info Plugin. From the Plugins top menu, choose “Leg # Master Configurator #”, then click “Configure”. Ensure the Legends Info Plugin is set to active.

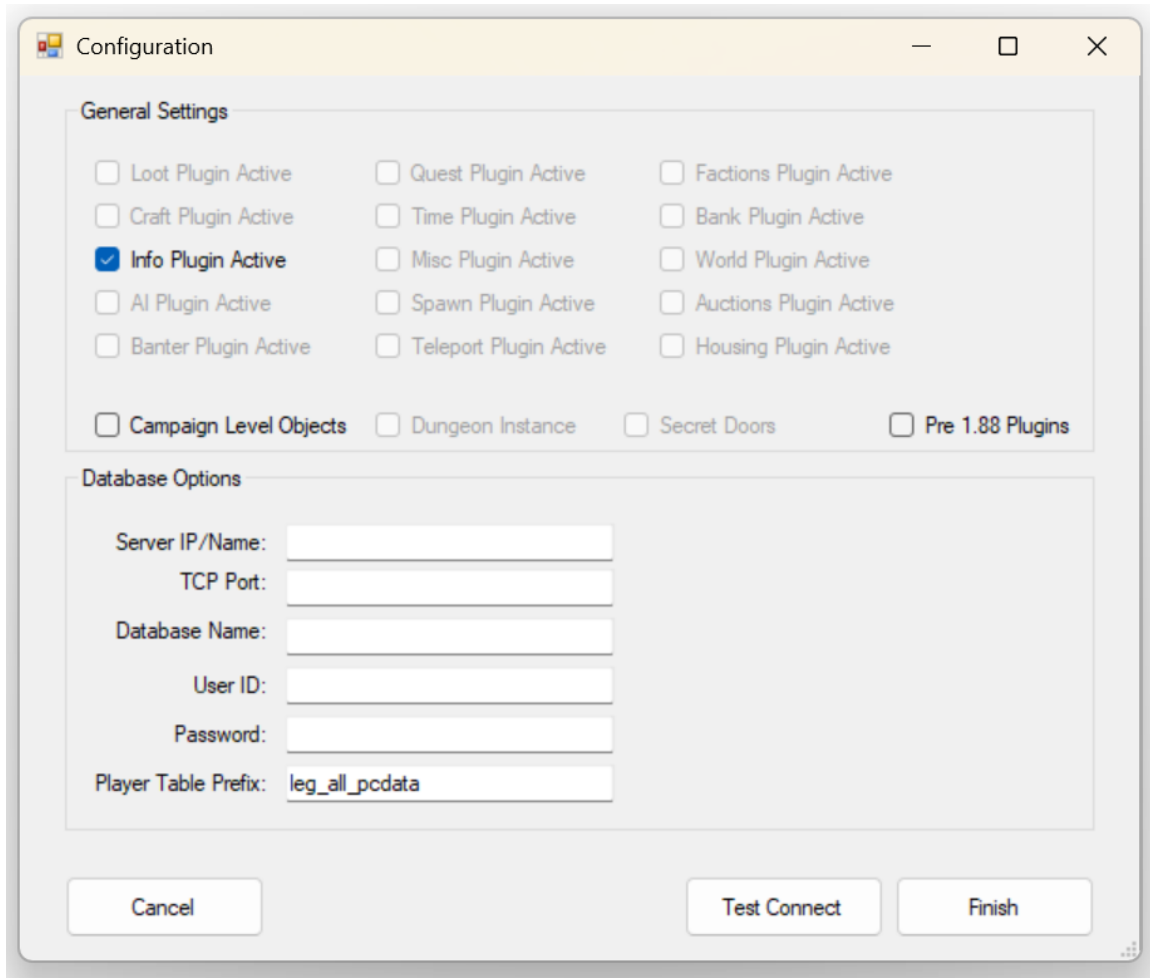


Figure 4 - Legends Master Configurator

- **CHECK BOXES** = These checkboxes are to activate and deactivate the various plugins. They will install any hakpak and set any module variables that are required.
- **CAMPAIGN LEVEL OBJECTS** = Selecting this causes the Legends plugin code to look for and use SCRIPTS in the campaign folder that is configured on your module. If you do not have a campaign configured and are not keeping scripts there, then do NOT select this option. Select this ONLY if you are keeping all your scripts in the campaign folder. NOTE: Importing ERF files causes scripts to end up in the module folder. You can manually move them to the campaign folder if you wish to go this route. Generated Master and Info constants are written to the selected script root. The leg_info.ini file stays in the module folder.

- **DATABASE OPTIONS** = Some Legends plugins require an NWNX and MySQL database backend. The Legends Info plugin DOES NOT require a database unless you intend to use the Town Crier features.
 - **PLAYER TABLE PREFIX** = This option is used to group any player created tables for ease of finding. Several Legends plugins will create tables for each player to track their game progress. Those tables will be prefixed by this entry. It is usually safe to leave this as is.
6. Once you've completed the configuration, click Finish and then note there are two other buttons available to use.

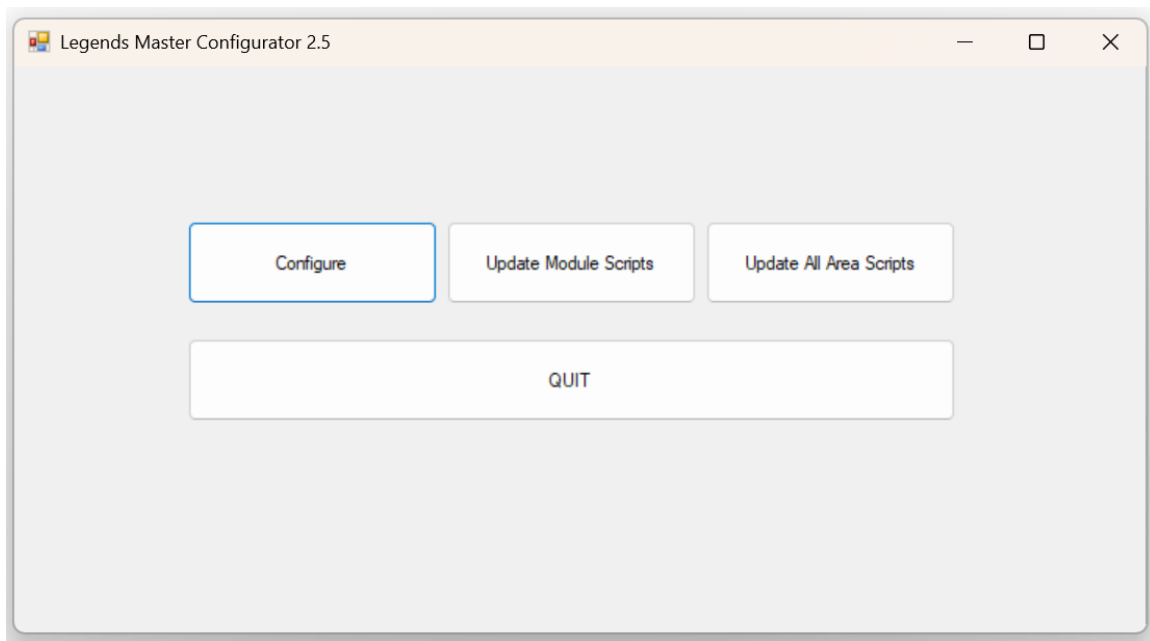


Figure 5 - Master Configurator

- **UPDATE MODULE SCRIPTS** = This button will update the module properties with all the appropriate module scripts with Legends Master scripts. The following events are updated. If an existing script is found in the required events, the Master Configurator will copy it, modify it, and place the new modified copy in place thus preserving your existing code functionality. If you wish to manually perform this step, you will need to ensure the following scripts are being called in the appropriate events. NOTE: Using this button will only place scripts in events that the currently active plugins require.
 - i. `leg_all_mastermod_itemacquired` – On Item Acquired Event
 - ii. `leg_all_mastermod_itemactivated` – On Item Activated Event

- iii. leg_all_mastermod_itemunacquire – On Item UnAcquire Event
 - iv. leg_all_mastermod_ondeth – On Player Death Event
 - v. leg_all_mastermod_onheartbeat – On Module Heartbeat Event
 - vi. leg_all_mastermod_onload – On Module Load Event
 - vii. leg_all_mastermod_onrest – On Player Rest Event
- UPDATE ALL AREA SCRIPTS = Use this only when the active plugin set requires the area OnClientEnter/OnExit integration. Back up the module first and review all generated or modified event-script calls.
7. Once you have finished configuring the Master Options, choose Plugins from the top menu and select “Legends Info Plugin”, then choose File -> Configuration.

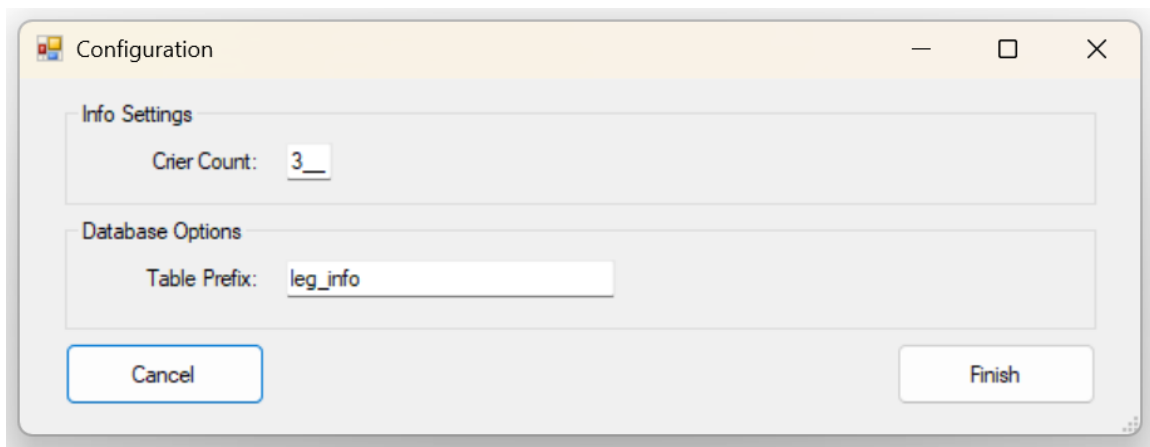


Figure 6 - Legends Info Plugin Configuration

- CRIERCOUNT = This is the total number of times a crier will speak each message in his message queue before moving on to the next message. A crier will only speak once per minute.
 - TABLE PREFIX = This is the prefix that you would like to use for Legends Info related database tables. Configuration checks the two Info tables independently, creates missing tables and conditionally renames existing tables when the configured prefix changes. Existing valid tables and settings can be retained.
- ⚠ Once complete, you must perform a Compile All Scripts in module from the File menu of the toolset or compile all of them with the compiler of your choice.

Part 2 – Making it Work

Tutorial 1 – Talk to me Goose

Now that you’ve completed the installation process, you’re probably itching to get started with the plugin. The first example of what we want to accomplish is to display some information on the adventurer’s screen. A common purpose for this is when someone enters an area or section of an area for the first time and you’d like to show the adventurer where they are.

For this tutorial, we’re going to use a special Info trigger that comes with this plugin. Create an area in your tutorial module or world, then select the “Legends Info Box” trigger in the Triggers group under Legends Info category. Once placed, select it in the toolset window, then choose Plugins -> Legends Info Plugin -> File -> Edit -> Trigger Instance.

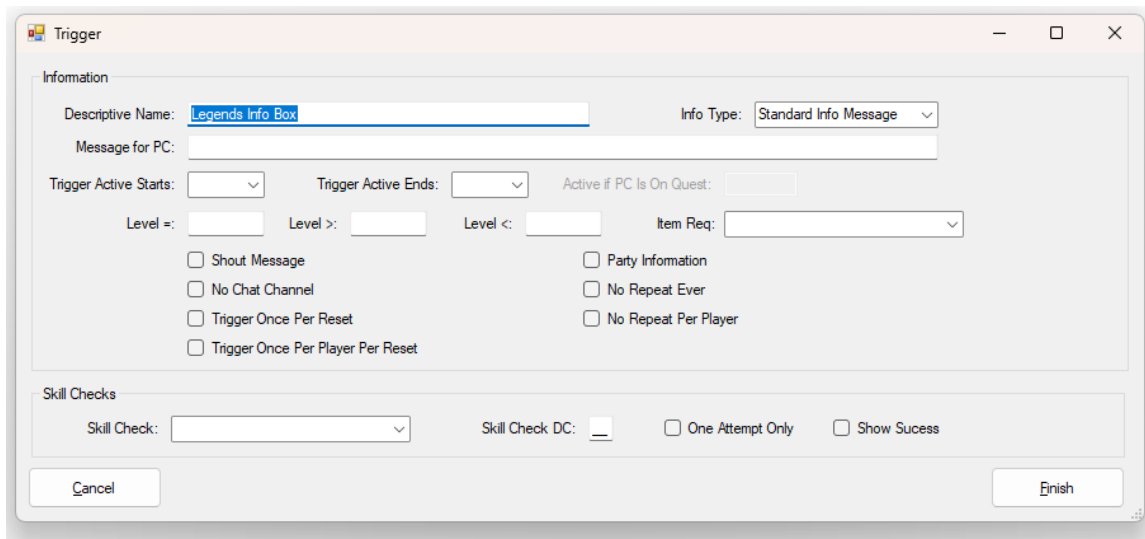


Figure 7 - Trigger Configuration

Descriptive Name This is an identifier for this trigger. It will help you see what it is easily when selected.

Message for PC This is the actual text the PC will see when they walk into this trigger.

In the Name field enter: “My First Trigger”.

In the Message for PC field enter: “What a nice grassy field.”

That’s it. Load up your module and try it out. Enter the area where you placed your info trigger and you should see your message appear on the screen.

Seems pretty simple, doesn't it? Let's move things along to make this Info behave differently.

Info Types

The standard info message that has been available in previous releases is a white text message in the middle of the screen. Since then, two new types have been added.

- Alert Info Message = This is the same as the standard info message however it appears in RED text and just below the standard info message. Both can appear at the same time.
- Detail Message Box = This is a multi-line text message box that appears on the right side of the screen like the Tutorial Message boxes seen in the official campaign.

All three types will use the "Message for PC" field for their content.

Tutorial 2 – Is it Just Me?

Sometimes when you trigger an event that causes Info to appear, you may want to share that information with your party, or even the entire server. Select your trigger again and choose Plugins -> Legends Info Plugin -> File -> Edit -> Trigger Instance.

Set your options as shown in the picture below.

Figure 8 - Configuring an Audience

Shout Message	Check this if you want the message shouted across the server.
No Chat Channel	Select this if you only want the text to appear in the PC's GUI window and NOT the chat dialog box.
Party Information	Checking this will cause the entire party to see the message if any one player should run over the trigger. Note that if each player runs over the trigger in the party with this on, it will result in spamming the party. It is advised to use this along with the Once Per Reset or No Repeat options.

Tutorial 3 – Taking Control

Having an info trigger message appear for every party member is quite helpful but sometimes sharing info like that can be quite noisy for everyone. You can limit this a little bit by adding the following checks that will help the situation. Once again, select your trigger and choose Plugins -> Legends Info Plugin -> File -> Edit -> Trigger Instance. Review the other checkbox properties:

Trigger Once Per Reset	Check this if you only want this trigger to show its message once per reset. This means that the first PC to trigger it will be the only one to see it until the server resets (unless configured for party info or shout).
Trigger Once Per Player	This checkbox allows each player to trigger the message per server reset once.
No Repeat Ever	This option allows the first person to trigger it and to never be active again.
No Repeat Per Player	This allows each player to trigger it once and once only.

Tutorial 4 – Various Checks

Time of Day

Sometimes we may only want a message appearing during certain times of day, such as during night or during day or even a message that appears at noon or midnight.

Adjusting these entries in the Trigger Instance window will have the following effects:

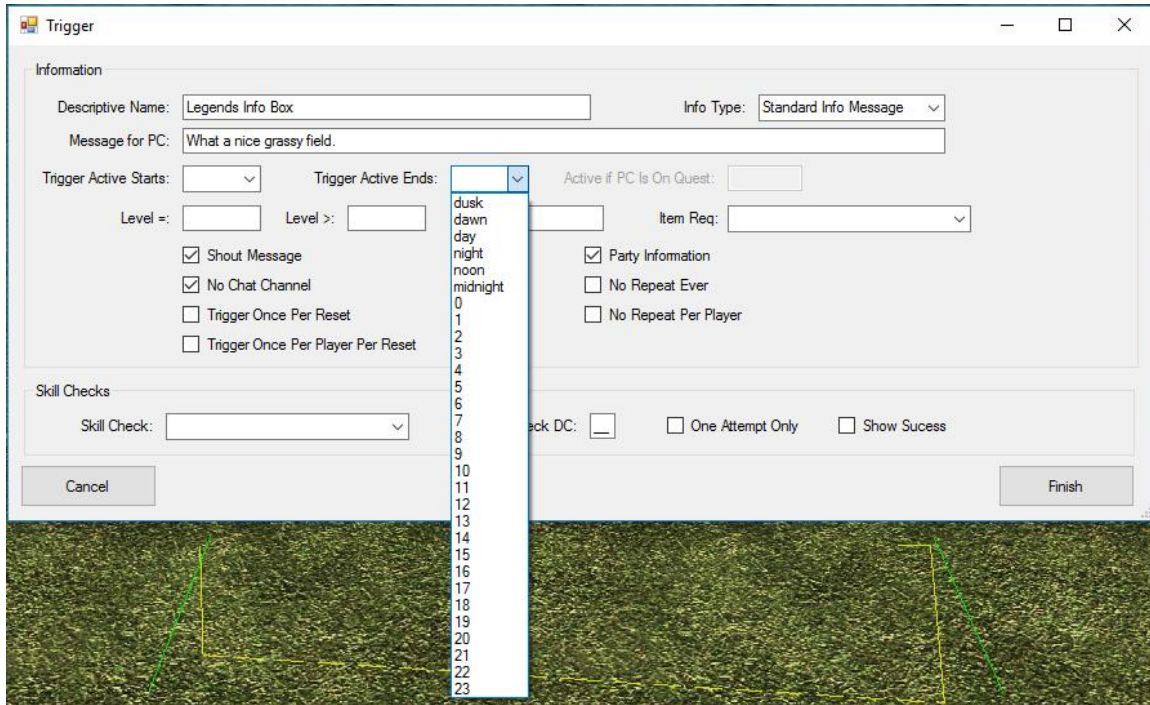


Figure 9 - Time of Day Settings

Trigger Active Starts This is the time that the trigger will become active.

Trigger Active Ends This is the time that the trigger will stop being active. If the start time begins at dusk for example, and the end time is set to dawn, the trigger will de-activate the following day.

Skill Checks

Optionally, you can assign a skill check to a trigger as well. This can be useful if you would like to have messages sent to players when they hear something, or Spot something. You can even set the “One Attempt Only” checkbox so that even if they fail, they won’t get repeated attempts.

Optionally, you can also check the “Show Success” checkbox so if a PC successfully triggers the Info Box with a skill check, it will show the player that a skill check was made and what the result and the DC was. If they fail the skill check, it will not show the player anything. This is useful for things like Spot and secret doors. Placing a trigger near a secret door and configuring it to use a Spot check can add a lot of substance.

Level Checks

You can use the three level fields to have the Info fire based on a player's level. You can make it fire for a specific level, if a player is below a certain level or if a player is above a certain level. You could even create overlapping triggers that would give a different message based on players with different levels.

Item Requirement

Sometimes it can be useful to trigger some information if a player has a particular item in their inventory. For example, if they are carrying a magical key, they could step on a trigger near an ancient door and have the info message comment that the key in their possession radiates magic near this door.

Tutorial 5 – Other Triggers

While one of the most common uses for info messages like these is on a trigger, it is also useful to have on creatures as well. A nice info message can optionally appear on the death of a creature if you wish. A placeable object when clicked on can offer up an info message as well.

For a placeable object, simply select one from an area (not a blueprint) and then choose Plugins -> Legends Info Plugin -> File -> Edit -> Placeable Instance.

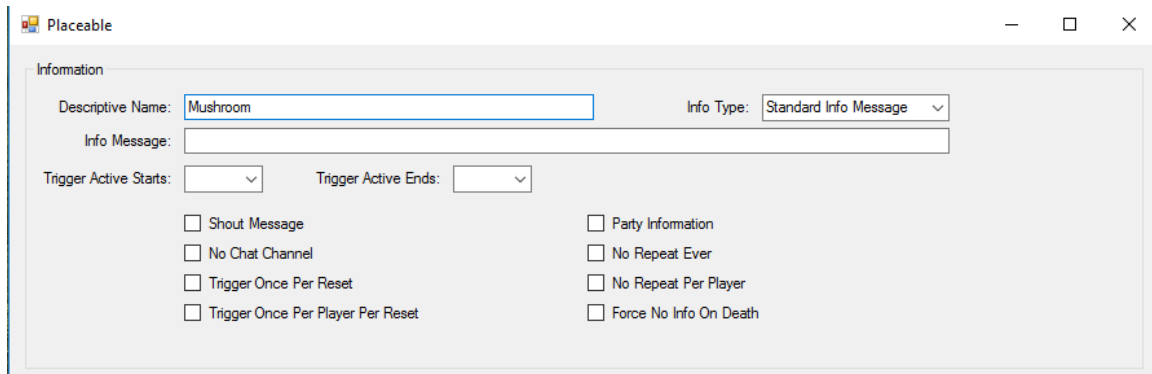
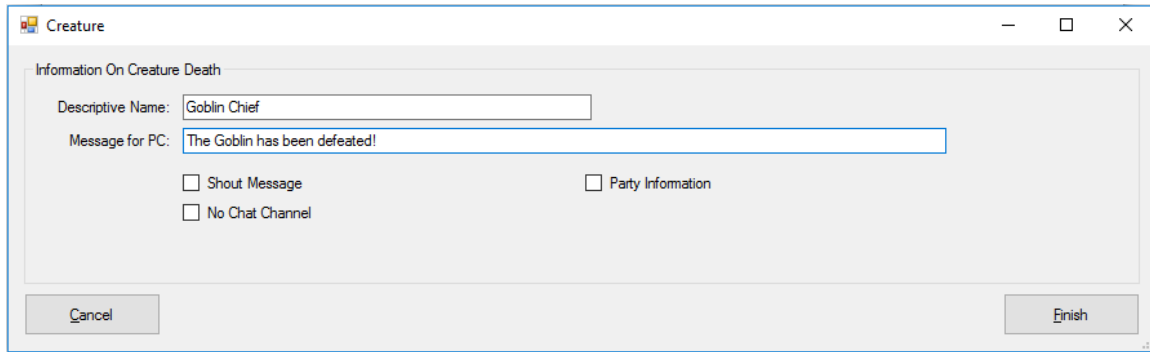


Figure 10 - Placeable Information

We will leave the Progress Bar trigger section of this placeable alone for this tutorial as those items will be described in a later tutorial. For now, we would simply like a message to appear when our placeable is clicked on. Using the same rules as our trigger, fill in the fields and click finish. Now whenever our placeable is clicked on, the message will appear just as if it were a trigger. All variables, scripts and other placeable settings will be configured automatically. Load up the module and click on the placeable!

If we would like to see a specific message appear when a creature is defeated, this can be accomplished easily.

For this tutorial, copy the blueprint of a goblin and change its name to “Goblin Chief”. Select the blueprint of our new Goblin Chief and then Plugins -> Legends Info Plugin -> File -> Edit -> Creature Blueprint.



Information On Creature Death

Descriptive Name:

Message for PC:

☐ Shout Message ☐ Party Information

☐ No Chat Channel

Figure 11 - Goblin Information on Death

These settings are the same as the other types of triggers so fill them in appropriately. Now place your new goblin chief and defeat him in battle. Upon his defeat, you will see the message entered here.

Tutorial 6 – Making Progress

Also included in the Info plugin is the ability to have progress bars. If you've ever performed a Rest action in NWN2, you may have noticed a small progress bar moving along the bottom of the screen. The progress bar doesn't do anything however it allows the scripter to create interesting time sinks for adventurers.

Consider this. You have cleverly written a script that will give the player an item when they click on a placeable object if they succeed in a search skill check. This is a great example of where the Progress Bar can come into play. By leveraging a progress bar with your script, you can cause the adventurer to appear to search the placeable object for several seconds, displaying your progress bar along the screen and then firing your script.

For this tutorial, we will combine both an info box from the previous tutorials and a progress bar to a placeable object. First, we set up our placeable object with some appropriate options. Choose a placeable object that you would like to have an adventurer search and place it in world. Select it, then fire up the Legends Info Plugin and select File -> Edit -> Placeable Instance. Enter the information in the top part of the window as in the previous tutorial. Next, we'll look at the bottom part of that window.

The screenshot shows the 'Placeable' configuration window. The 'Information' section at the top has 'Descriptive Name' set to 'Mushroom {02 {x1}}' and 'Info Type' set to 'Standard Info Message'. Below this are fields for 'Info Message', 'Trigger Active Starts', and 'Trigger Active Ends'. A group of checkboxes includes 'Shout Message', 'No Chat Channel', 'Trigger Once Per Reset', 'Trigger Once Per Player Per Reset', 'Party Information', 'No Repeat Ever', 'No Repeat Per Player', and 'Force No Info On Death'. The 'Progress Bar Trigger' section below contains various fields and checkboxes: 'Progress Bar Text', 'Success Script', 'Fail Script', 'Timer Fail Script', 'Skill Check' (dropdown), 'Equipped Item Req' (dropdown), 'Skill Message', 'Animation' (dropdown), 'Visual Effect' (dropdown), 'Progress Bar Duration', 'Timer Bar Duration', 'Set SE Flag' (dropdown), 'Skill Check DC', 'Item Req' (dropdown), 'Give Item' (dropdown), 'Skill Show', 'Show Skill Success', 'PC Can Move', 'One Attempt Only', 'Call Script Deactivates', 'Use the Trigger's Name In Progress Bar Text', 'Can Cancel', 'Remove Item On Use (On Success)', 'Looping Progress Bar', and 'Apply Effect to PC'. 'Cancel' and 'Finish' buttons are at the bottom.

Figure 12 - Progress Bar Configuration

As you can see, progress bars can have a lot of options assigned to them. Let's review them.

Progress Bar Text	This is the text of what will appear in the Progress Bar GUI. You can put “Searching Tracks”, “Looting Keg” or anything that you like to have shown in the text area of the bar.
Success Script	This is the name of a script that will fire if the player uses a skill check and is successful and the progress bar has completed. This is where you would place your custom script for a successful search for example. Also, this field is used as the default script to execute if there is NO skill check configured. (Essentially automatic success). If no script is entered here, the progress bar will complete, and the Info message will be displayed with no further actions.
Fail Script	This is a script that can optionally be fired if a skill check is failed. Sometimes you may want something special happen such as a trap trigger or other function that you as a scripter can provide should the player fail their check.
Timer Fail Script	Timer progress bars work a little differently in that they count down to something. Should a progress bar be configured as a Timer Bar, the player is given an amount of time to complete a task and shut the bar off. If the bar should be allowed to complete all the way to the end, this script will fire.
Skill Check	This is a skill check that can be configured giving the progress bar more usefulness. Searching, Sleight of Hand, Survival are all useful skills that can be used here from looking for an item, to lighting a campfire.
Equipped Item Requirement	This is a check to ensure the player has an appropriate item in their hand before the progress bar is allowed to begin. Perhaps they need a certain piece of equipment?
Skill Fail Message	This is an alternate message that can be displayed when the progress bar completes and a skill check has failed. Normally when the progress bar completes, the “Info Message” text is displayed. If there is anything in this field, it will be displayed instead on skill failure.
Prog Bar Duration	This is how many seconds the progress bar will take to complete.

Timer Bar Duration	This is how many seconds the timer bar will take to complete. More on Timer Bars later in this document.
Set SE Flag	This is a tie into the Legends Special Effects Plugin which comes packaged with the Legends Quest plugin. Should a progress bar be successfully completed, a Special Effects flag can be set and reactions occur based on that plugin. See the Legends Quest plugin documentation for more on Special Effects flags.
Skill Check DC	If a skill check has been selected for this placeable progress bar, this is the difficulty value for the check.
Item Req	Similar to Equipped Item Requirement, this is an item that must be in the player's inventory for the progress bar to activate.
Give Item	On a successful progress bar completion, this can be an item the player will receive. Using this can make simple item searches easier and does not require a success script to only hand out an item.
Skill Show	Having this flag set will display information in the adventurer's chat window about the skill roll that is occurring and the results. Without this, the skill check is made silently.
PC Can Move	Setting this checkbox will allow an adventurer to still move around and interact during the progress bar.
One Attempt Only	This option will cause players that fail a skill check to no longer be able to attempt to use the object. Be careful with this one; if a player requires a successful skill check to progress a story or quest, failing with this option active, could cause a road block.
Call Script Deactivates	When a placeable object is in use and a progress bar is running for a player, clicking the placeable again will cause an "Object Busy" message to appear. By default, when the progress bar is complete, the placeable object becomes free to click on once again. Sometimes however you do not want the placeable object to be clickable until something else happens. When this checkbox is on, the info plugin will not release the placeable object and expects a follow up script to do it.

For scripters, to manually make a placeable useable again in your custom script, you need to set the following variable:

```
SetLocalInt(OBJECT_SELF, "LEG_INFO_IsActive", 0);
```

Where OBJECT_SELF is the placeable.

Use the Trigger's Name

When checked, the Progress Bar will append the NAME field of the placeable object to the text in the Progress Bar Title. For example, by placing the word "Examining" by itself in the Title and then setting this on a Pile of Rocks placeable would create a progress bar with the words "Examining Pile of Rocks" on the screen.

Can Cancel

If this checkbox is active, the progress bar will appear with a Cancel button on it. Pressing the cancel button will abort the progress bar and stop any script that may have fired at the end of it.

Remove Item On Use

This feature when checked will remove any item specified in the "Item Req" field.

Looping Progress Bar

When this checkbox is turned on, the progress bar becomes a Looping Progress Bar. This means the progress bar will fire, run its follow up scripts, and then start again. Each time the bar reaches the end, it will re-fire its scripts. The bar will also have a Cancel button on it so that a player can stop the loop when they desire.

The other items in this configuration window are optional and are described in the next tutorial.

Tutorial 7 – Skilled Progress

Often we would like to leverage our character skills a little more. A progress bar is a great way to make use of a wide variety of them from including Search, Spot, Lore, Survival and many other skills.

Consider this. You have written a custom script to have an adventurer light a fire on a campsite placeable and stay lit for 30 seconds when it's clicked on. Leveraging a progress bar with skill support, you can have the adventurer click the fire pit, cause them to wait 10 seconds while they attempt to light the campfire, make a roll against their survival skills and fire an appropriate script should they succeed.

For this tutorial, we're going to do just that. First, you need to create a custom script for actually making fire appear on the campfire placeable object you will be placing in a moment. Create a new script called "mycampfire" and add the following code to it:

```
// Little script to make fire appear on a placeable

void main()
{
    float fTime = 30.0;
    effect eCampfire = EffectNWN2SpecialEffectFile("fx_torchglow.sef");
    ApplyEffectToObject(DURATION_TYPE_TEMPORARY, eCampfire, OBJECT_SELF, fTime);
}
```

Next, place a campfire placeable on the ground in world. Make these changes to the placeable using the Legends Info Plugin. Choose File -> Edit -> Placeable Instance.

Figure 13 - The Campfire Placeable

Now run your tutorial module and click on the little fire pit placeable object. If you succeed in your skill check, your custom script will fire. If you fail, the info message will appear on your screen.

Skill Check

This is the NWN2 toolset skill you wish to check. In this example we are using the skill called Survival.

Skill Check DC	This is the difficulty of the skill check. Using this value, we roll against our skill to determine if we are successful or not.
Skill Fail Message	This is the info message that will appear on the adventurer's screen should they fail their skill roll. Leaving this blank will not display any message.
Skill Show	Having this flag set will display information in the adventurer's chat window about the skill roll that is occurring and the results. Without this, the skill check is made silently.
Fail Script	Similar to the Execute Script option above, this is another script that you can optionally call when a skill check fails. We'll leave this blank for this tutorial.
Animation	On progress bars, the PC will normally crouch and perform an action while the progress bar ticks however you can change this behavior by specifying an alternate animation here. We'll leave this for this tutorial.
Visual Effect	The visual effect is useful for more magical type progress bars where the player may be conjuring something or summoning something. You can select a visual effect from the list here.
Apply Effect to PC	By default the visual effect above appears on the object the player is interacting with however you can override that behavior and have the effect appear on the player themselves instead by using this checkbox.

Tutorial 8 – Lights, Camera, Action!

When an adventurer enters a new town for the first time, it's sometimes nice to give them a little information in game about the place, saving them from having to refer to some document on a web site if they just want some brief information about it. Sometimes, a revelation may occur that is so important that only a sequence of images and scenes can properly relay this information. All this capability is already built in to the NWN2 toolset however this plugin and tutorial helps make it a little easier to create simple cut scenes for your adventurers.

First, let's imagine we have a town created and we would like to display some information about the town when the player first arrives. To do this, we start by deciding what information we want to tell them. For this example, we want to let the adventurer know that this is a quiet town, but they seem to be experiencing some trouble with the local wildlife. For this, we will use 3 lines of text.

Step 1 of this tutorial is to place the special Info Plugin camera trigger which can be found in the Legends Info category in Triggers. Put the trigger somewhere near the place where the adventurer enters the area, but not right on the spot where they enter. We'd like them to be fully loaded into the area and must take a step or two before our scene activates.

Once you have it placed in world, select it, fire up the Legends Plugin and select File -> Edit -> Camera Cutscene

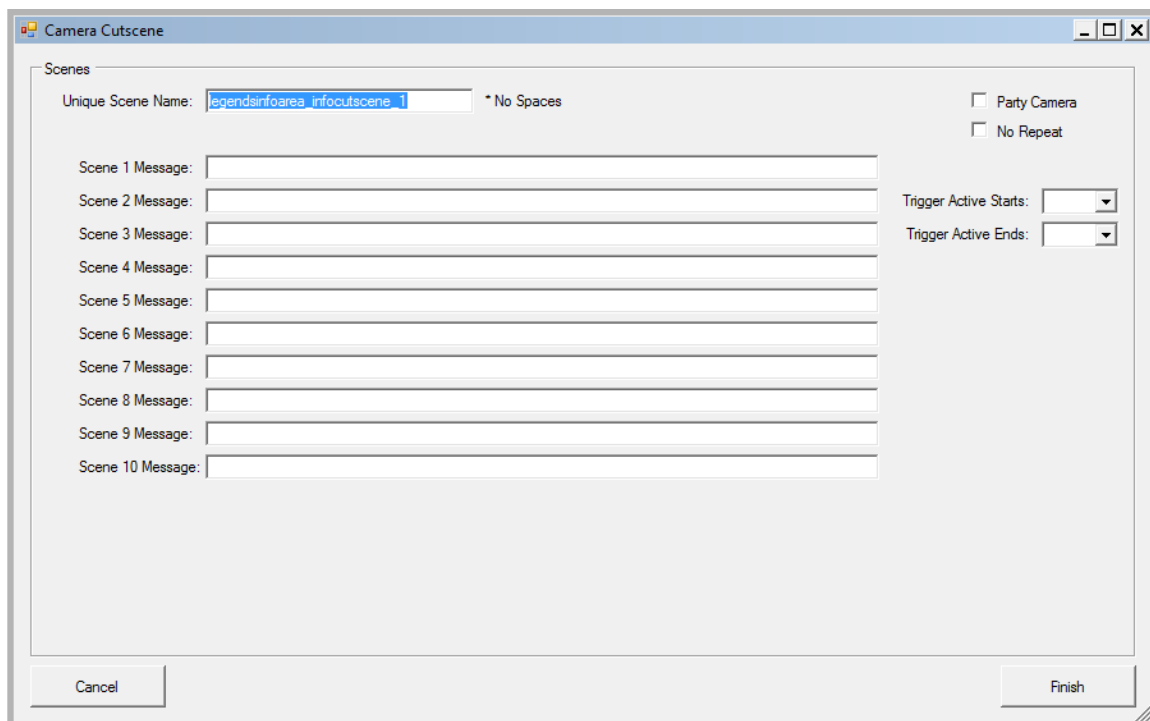


Figure 14 - Camera Cutscene Configuration

These options for our cutscene are described as follows:

Unique Scene Name	This is the name of the cut scene and needs to be unique. It is also the name of the conversation that is created and used in several other places. Do not use any special keyboard characters for this field. A good naming convention is the areatag_<unique number> for the area.
Party Camera	Checking this will cause everyone in the party to see the same cut scene when the first person triggers it. Without this variable, each party member would have to cross the trigger individually to see the scene. Only party members that are in the same area as the adventurer that triggered the scene will see it.
No Repeat	Checking this will make this cut scene only fire once for each adventurer. Once it has completed, it will not play again for them even after a server reset.

Note that the Trigger Active Starts and Ends also work on cut scene triggers so you can have them activated only during certain times of the day, just like standard Info messages from the earlier tutorials.

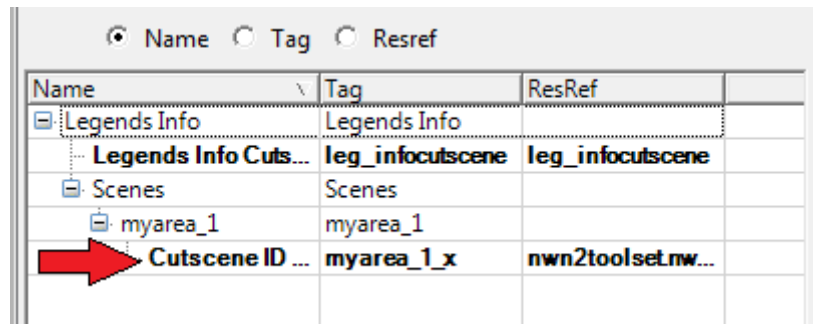
You can have up to 10 different scenes or camera angles in your cutscene. Each scene will have some message text that will be displayed. You can use the 10 message boxes in the configuration window to enter the text you'd like to appear for each scene. Each message appears in order starting with number 1 from our scene. For each message box used, we will want 1 camera angle to go along with it. For this tutorial, let's make three scenes using Scene Messages 1-3 enter the following:

“You arrive at a quiet little town.”

“Things seem peaceful here, but that is only your first impression.”

“You will soon learn of the troubles this town has with the local wildlife.”

Click Finish. At this point several things will happen. The most important is that a new camera will be created in your blueprints under Legends Info -> Scenes -> Scene Name.



Name	Tag	ResRef
Legends Info	Legends Info	
Legends Info Cuts...	leg_infocutscene	leg_infocutscene
Scenes	Scenes	
myarea_1	myarea_1	
Cutscape ID ...	myarea_1_x	nwn2toolset.nw...

Figure 15 - New Camera Created

Because we are using 3 scene messages, we will need to place this camera down in our area 3 times. Once we have done that, select one of them and place it into position for scene 1. It is best to have the Preview window open when positioning your camera as it's easier to see what will be in the scene:

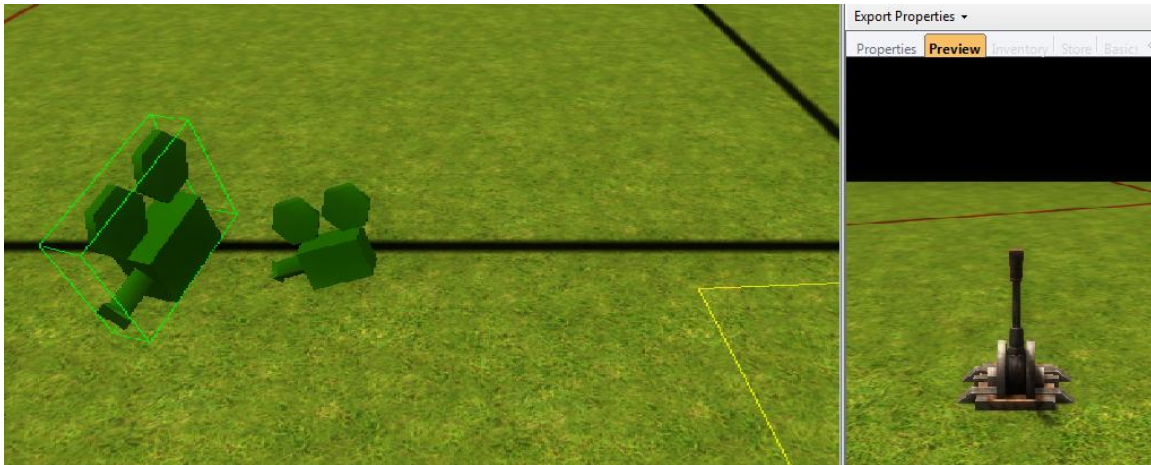


Figure 16 - Using Preview to See the Cam

Once you've placed all three cameras down, and positioned them, change their TAG. You will notice each tag is already configured and ending with an "x". Simply change this x to match the scene number. So, for scene 1, change the x to a 1 and so on. If you named your scene "myarea_1" then the tag for the scene 1 camera will read "myarea_1_1".

You are now complete with the cutscene. Enter the game and walk over the trigger.

Tutorial 9 – Hear ye, Hear ye!

Another handy method of relaying information to adventurers is with ye'ol town crier. The Legends Info Plugin supports the ability for you to place an NPC and have him or her speak messages that have been applied.

A special script called “leg_info_crieradd” allows you to inject messages into your world's crier's queues. You can inject as many as you wish, as often as you wish and the crier will speak them in order until either the server is reset or they have gone through all of the messages in their queue where they will wait until the next injection of messages.

You can have as many criers as you like in your world, and you can specify to inject messages into a particular one or all of them. This is a handy feature when you would like to announce an event such as the Goblin Chief being killed, or an item plundered from the town vault. Pretty much anything goes. The trick to the crier is calling the leg_info_crieradd script when an in-world event occurs.

Each crier must have an associated placeable object to store their message queue as the Crier itself may not be always spawned. The placeable object can be anything and often placed near the spawn point of the crier. The Legends Info Plugin will create an invisible placeable object automatically for you however you can change its appearance as you see fit. It can be a banner, a torch, anything; as long as it is not static. In this tutorial, we will go through the creation of our crier, his placeable message queue holder and a monster for us to slay and thus triggering the crier. The number of times the crier will speak each message is configured during configuration of the plugin above.

Each Crier message can contain up to two lines. Commonly, the first line could be used to tell when the event happened, the second, what the event is.

First, choose an NPC to act as a crier from the creature blueprints. This must be a copy of one, or a custom created one as built in creatures cannot be modified.

Next open the Legends Plugin and select File -> Edit -> Crier and Queue

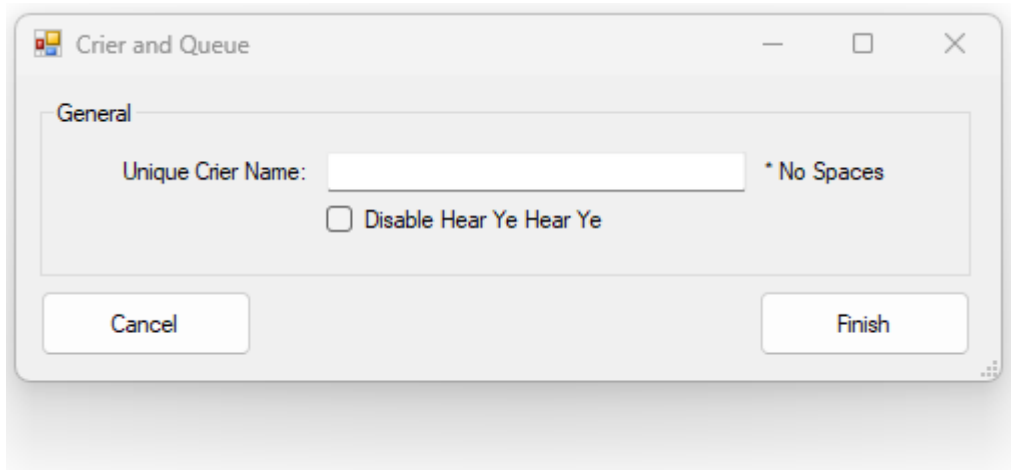


Figure 17 - Creating a Crier

Unique Crier Name

This is the unique ID of this crier's message queue placeable. It is also stored in the leg_info_criers database table. Do not use spaces or any special keyboard characters.

Disable Hear Ye Hear Ye

Normally, a crier will declare "Hear Ye, Hear Ye" before announcing your messages. Checking this box will disable that.

Place your crier in world and look in the placeables blueprints where you will find a matching placeable object for your crier in the Legends Info -> Crier Queues -> Crier ID

Name	Tag	ResRef
Legends Info	Legends Info	
Crier Queues	Crier Queues	
mycrier_1	mycrier_1	
Legends Cri...	mycrier_1	mycrier_1

Figure 18 - Crier Queue Placeable

Place this in world near your crier. It will have an invisible appearance however you can adjust this to be anything you like if you prefer.

Your crier is now ready to start speaking messages. All you need to do now is inject one. We can do this easy for many Legends type events such as the death of a monster. For your own custom events, you simply need to add the following to your script:

```
AddScriptParameterString("First line of my message.");
AddScriptParameterString("Second line of my message.");
AddScriptParameterString("");
ExecuteScriptEnhanced("leg_info_crieradd", OBJECT_SELF);
```

Note, that if you want only a specific crier to speak this message, change the third parameter string to the Unique Name of the crier's message queue placeable object. In this case, we are going to use a Legends event on a monster when it dies. Choose a goblin and place it somewhere for you to fight. Change its name to "The Mighty Goblin". Modify its on death event script, or make a copy if using the default NWN2 scripts and add the following code near the bottom:

```
ExecuteScript("leg_all_masterdeath", OBJECT_SELF);
```

NOTE: If using the Legends AI Plugin with this goblin, this will already be done for you.

Next, add the following variable to the goblin:

Variable Name	Type	Value
LEG_INFO_CrierAddMyDeath	Int	1

Figure 19 – Goblin Chief Variables

LEG_INFO_CrierAddMyDeath	When set to true, this will announce the death of the creature it is configured on and the person who struck the final blow. The message will be spoken to all configured criers.
--------------------------	---

Now to test this tutorial, you will need to run the module using NWNX. Try it out and see what happens. Defeat the Mighty Goblin and then move to where the crier is.

Final Summary

This system was originally designed for the NWN 2 Persistent World known as Legends. The plugin outlined here was designed as built into the Legends Quest systems however it was deemed valuable enough to extract into its own stand-alone plugin.

For support and current releases, visit <https://nwn2legends.com/>.

I hope you enjoy this system and feel free to season to your individual needs and tastes!
Have Fun and Happy Info!

© 2026 Marshall Vyper. Legends Info Plugin and documentation.